```
          *************************************************************
                                         ***
                  /////////      ////   ////  ***  /////////
                 IIIIIIIII/     IIII/  IIII/    ***IIIIIIII//
                IIIII/          IIII/  IIII/     IIII/  IIII/
                IIIII/          IIII/  IIII/     IIII/* IIII/
                IIIII/          IIII/  IIII/     IIII/***III/
                IIIII//////  // IIII///IIII/  // IIII///***I/  //
                 IIIIIIIII/ II/  IIIIIIIII/  II/  IIIIIIII*** II/
                                                          ***
          *************************************************************
                  CIRCULO   DE    USUARIOS    DE      QL
          *************************************************************
```

Fanzine mensual independiente para usuarios de Sinclair QL y compatibles

---

AÑO 2 NUMERO 9 JUNIO DE 1989

---

...en el año 9 Después de Sinclair, toda Hispania está ocupada por las legiones de PCs. ¿Toda? ¡No! Un puñado de irreductibles QLs resiste todavía y siempre ante el primitivo invasor...

---

CONTENIDO

---



---

INFORMACION SOBRE C.U.Q.

---


Compilación de colaboraciones: Salvador Merino
Títulos y texto "Información sobre C.U.Q.": Marcos Cruz
Sub-Distribución: Marcos Cruz
Imagen de la portada: PANTALLA DOMINATION (PDQL)

SE CONSIENTE LA REPRODUCCION TOTAL O PARCIAL DEL CONTENIDO DEL FANZINE, PARA USO CULTURAL Y NO COMERCIAL, POR CUALQUIER MEDIO FISICO, QUIMICO, OPTICO, MAGNETICO, SOLAR, MECANICO, HIDRAULICO, EOLICO, ELECTRICO, NUCLEAR O A PEDALES, SIEMPRE Y CUANDO SE CITE LA PROCEDENCIA (¿EH? ¡OJO!).

El fanzine C.U.Q. es un medio de comunicación e intercambio de ideas y conocimientos entre usuarios de ordenadores Sinclair QL y compatibles. El fanzine se realiza mediante la colaboración desinteresada de todas las personas que lo desean.

¿COMO SE PUEDE RECIBIR EL FANZINE?

El fanzine se envía a quien contribuye a su realización aportando alguna colaboración, o bien a quien lo solicita sin más.

Quienes sólo quieran recibir el fanzine, sin enviar ninguna colaboración para que sea incluida en él, deberán enviar UN disco formateado de 3.5 pulgadas o DOS microdrives formateados, junto con un sobre franquedo y con las señas escritas para el reenvío. Estos discos o microdrives SIN COLABORACIONES deben enviarse a: Marcos Cruz, Acacias 44, 28023 Madrid

El fanzine de cada mes estará disponible, para ser enviado a quien lo solicite, a finales de la primera semana de dicho mes, con lo cual pueden enviarse las peticiones a comienzos de mes.

¿COMO SE PUEDE COLABORAR EN LA REALIZACION DEL FANZINE?

Quien desee colaborar en la realización del fanzine, debe igualmente enviar UN disco de 3.5 pulgadas o DOS cartuchos de microdrive (aunque sólo uno contenga algo) junto con un sobre franquedao y con la dirección escrita para el reenvío. Estos discos o microdrives CON COLABORACIONES deben enviarse a: Salvador Merino, Ctra. Cádiz (Cerámicas Mary), 29640 FUENGIROLA (Torreblanca del Sol), MALAGA

¿Qué puede enviarse como colaboración? Cualquier cosa: artículos, cartas, opiniones, preguntas, respuestas, ideas, sugerencias... Si bien el nexo de unión entre quienes hacen posible C.U.Q. es el interés y la inquietud por el ordenador Sinclair QL, y por la informática en general, y esa es la temática global del contenido del fanzine, C.U.Q. acepta artículos sobre cualquier tema que pueda interesar a los lectores.

Con todas las colaboraciones recibidas a lo largo de un mes, se confecciona el fanzine correspondiente al mes siguiente. Los discos y microdrives con los boletines se echarán al correo en el último fin de semana del mes.

No deben enviarse más colaboraciones hasta haber recibido de vuelta los discos o microdrives con el fanzine del mes correspondiente.

FORMATO DE LAS COLABORACIONES

Los textos deben ser ficheros _doc del programa Quill, con los siguientes márgenes: izquierdo=1, derecho=80, sangrado=5. Los textos deben ir justificados a la derecha, y no deben emplearse caracteres no castellanos cuyo código ASCII sea mayor de 127. Tampoco deben usarse los corchetes cuadrados, ni las llaves, ni las barras vertical o inclinada hacia la izquierda.

Los artículos deben agruparse temáticamente según las siguientes secciones:

| | |
|---|---|
| CAR | Cartas abiertas |
| NOT | Noticias |
| LIB | Comentarios de libros y otras publicaciones |
| PRO | Comentarios de programas |
| HAR | Comentarios sobre hardware, y montajes |
| BAS | Lenguaje SuperBASIC |
| FTH | Lenguaje Forth |
| LEC | Lenguaje C |
| ASM | Lenguaje ensamblador 68000 |
| COB | Lenguaje COBOL |
| OFE | Ofertas, compra-venta, distribuidores, anuncios |
| OTR | Otros temas de interés, aunque no sean infomáticos. |

(Esta lista podrá ser ampliada o modificada por el recopilador, a sugerencia de los colaboradores.)

En el caso de que la colaboración consista en preguntas, o respuestas, deberá igualmente incluirse en una de las secciones indicadas, según el tema de a que se refieran.

Todas los artículos que pertenezcan a una misma sección deben ir en un mismo fichero. El nombre de este fichero tendrá el formato siguiente: SSSxxnnn_doc, donde "SSS" son los tres caracteres correspondientes a la sección de destino del texto, "xx" es el número de fanzine al que se envía el texto y "nnn" son las iniciales del autor.

En el texto, el título de cada artículo deberá ir en mayúsculas, en letra negrita, y a la izquierda de la línea (en el margen de sangrado). Excepto la

letra negrita en los títulos de los artículos, no deberá emplearse ningún tipo de letra (subrayada, negrita, subíndice o superíndice) en los textos.
Deberá dejarse siempre UNA SOLA línea en blanco para separar el título de un artículo del artículo anterior, si lo hay. Dentro de cada artículo deberá evitarse por completo el incluir líneas en blanco, tanto para separar párrafos como por cualquier otro motivo, con el objeto de ahorrar espacio en las copias impresas.
Al final de cada cada artículo debe ir el nombre del autor, en la columna 40, seguido en la misma línea por la localidad y la fecha.
Algunas importantes recomendaciones a observar:
-Déjese siempre un espacio detrás de cada signo de puntuación (coma, punto, punto y coma, dos puntos) ¡pero no delante!
-No deben aislarse con espacios los signos de interrogación, admiración, guiones o paréntesis, sino que deben estar unidos a la palabra siguiente -si son signos de apertura- o anterior -sin son de cierre.
-Cuídense tanto la ortografía y las tildes como los errores de "tecleo". Para ello, es recomendable releer lo que se ha escrito.

¿EN QUE FORMATO SE RECIBE EL FANZINE?
El fanzine se recibe en forma de varios ficheros _doc del procesador Quill, al objeto de permitir el acceso al mísmo de los usuarios que no tienen unidades de disco y/o ampliación de memoria.
El nombre de cada fichero indica el número de fanzine al que pertenece y la sección que contiene.
Para componer el fanzine debe cargarse el fichero CUQxxPOR (donde xx es el número de fanzine) y seguidamente deben UNIRse al final del texto, uno a uno, los ficheros de las restantes secciones en el orden que indique la columna "Sección" del índice del fanzine. Seguidamente puede alterarse el número de líneas por página, asignarse un pie de página y/o un encabezamiento y, si se desea, añadir cambios de página forzados para evitar que los títulos de secciones o artículos queden al final de alguna página. Finalmente, se ha de grabar el fanzine completo en disco y/o sacar una copia por impresora.
Si, por falta de memoria, no puede componerse todo el fanzine entero, deberá imprimirse por partes, a juicio de cada uno.
Una vez tenemos el fanzine en papel, habrá que rellenar a mano los números de página del índice.

¿PUEDEN INCLUIRSE DIBUJOS EN EL FAZINE?
Normalmente la portada llevará un dibujo (preferiblemente relacionado con alguno de los artículos del fanzine). El dibujo será un fichero de pantalla, en modo 4 u 8. El nombre de estos ficheros debe ser: DBPxxnnn_MODEm, donde "m" es el modo de pantalla en el que están realizados. Como en los textos, "xx" es el número del fanzine de destino, y "nnn" son las iniciales del autor.
El dibujo elegido por el recopilador para ser la portada del mes, se incluirá en el fanzine con el nombre CUQxxDBP_MODEm (donde "xx" es el número de fanzine y "m" indica el modo de pantalla, 4 u 8). Este fichero deberá emplearse para sacar un "hard-copy" sobre el folio de la portada.
La inclusión de dibujos en los artículos aún no es posible, pero están en estudio varios métodos posibles.

¿COMO ENVIAR PROGRAMAS AL FANZINE?
El fanzine C.U.Q. sirve también como intercambio de programas (PROGRAMAS REALIZADOS POR LOS MISMOS COLABORADORES, NO PROGRAMAS COMERCIALES). Estos programas pueden estar realizados en cualquier lenguaje disponible para el QL. En la medida de lo posible, y siempre que su longitud no sea excesiva, los textos fuente de dichos programas deben incluirse en un artículo (en la sección del lenguaje correspondiente) en el que se explique el funcionamiento u otros detalles que el autor considere oportuno. Los listados de programas que se incluyan en el boletín, no deberán tener líneas con más de 80 caracteres. Si el autor prefiere no ceder al fanzine el texto fuente, deberá igualmente escribir las instrucciones de manejo en un pequeño artículo, para su inclusión en el fanzine. Los nombres de los ficheros de que conste el programa deberán estar provistos de un nombre o prefijo idéntico para todos, de modo que sean fácilmente manipulables mediante los comandos WDIR y WCOPY así como de un sufijo (extensión) aclaratorio sobre el tipo de fichero (por ejemplo _task, _exe, _bas, _fth, _bin...)
El fanzine C.U.Q. no mantiene una librería de programas como tal, puesto que, por enviarse éstos en el mismo disco del fanzine, no es necesaria. No obstante, si alguien precisa alguno de los programas, puede solicitarlo. La lista de los programas que se han enviado junto con cada número de CUQ,

aparecerá en las últimas páginas de cada fanzine, con el nombre de su autor y una breve descripción de su función y características.

PARA MAS DETALLES SOBRE ENVIO Y PETICION DE PROGRAMAS, VER SECCION CUQ SOFT.

----------------------------------------------------------------------

## EDITORIAL

----------------------------------------------------------------------

Este número es sin lugar a dudas el más gordo que hemos hecho entre todos hasta hoy, pero mucho me temo que nuestro reducido grupo no puede permitirse el lujo de hacer un esfuerzo tan descomunal todos lo meses. Aún asi, tenemos un buen stock de material para ir tirando este verano, pero nuestro primer objetivo es llegar a fin de año (aunque tenga que escribir CUQ yo solito).

He recibido una carta de un usuario de Murcia y dos de Las Canarias que han visto el anuncio CUQ en Ordenador Personal.

Serafín me ha confesado que unos 100 socios antiguos de QLave eran de CANARIAS. Si eso es verdad, significa que donde más socios habia ni se conocian, o no interesaba que se enterasen. La verdad es que pocos Canarios han participado escribiendo algo para QLave.

S. MERINO

----------------------------------------------------------------------

## CARTAS ABIERTAS

----------------------------------------------------------------------

NOTA: Las cartas dirigidas a esta sección deben ir tituladas con una breve frase que resuma lo que se quiere expresar en ellas.

[[[ ESO ES PARA QUE SEA MAS FACIL VER DE UN VISTAZO LOS TEMAS DE LOS QUE SE TRATA, ASI COMO PODER MENCIONAR EN EL INDICE LAS CARTAS INDIVIDUALMENTE ]]]]

UN EMULADOR DE QL PARA PC

Hola Salvador:

Te mando una colaboración para el boletín.

El tema a tratar va a ser los programas de arquitectura que lanzó Investronica para el QL. Pero antes contaré el final de la historia. Me voy a comprar un PC para trabajar profesionalmente en él, y seguiré enredando con el QL. La compra de un PC es algo muy complicado, porque todo son tarjetas y trucos para mantener la compatibilidad. Me he decidido por un XT con disco duro. Eso si, hay programas para todos los gustos, y muchos traducidos al castellano.

Ahora lo que mas me interesa es ver que posibilidades de conexión hay entre un XT y el QL. Puedo utilizar los discos con el QL? puedo conectar los ordenadores por la puerta serie?

Mi problema es que funciono con mdvs. y no tengo discos en el QL. La idea que tenía era utilizar el paquete Psion en el PC y utilizar el QL como unidad portatil. ¿Que nombre tienen en el mundo PC los cuatro programas?

ME HA LLEGADO INFORMACION SOBRE UN PAQUETE QUE EMULA AL QL EN UN PC. Cuando sepa algo más ya lo contaré. (Salvador seguro que se muere de risa.)

Respecto de los programas de arquitectura, yo he visto tres:

Cálculo de vigas trianguladas, Cálculo de estructuras metálicas, y el paquete de mediciones y presupuestos.

En conjunto, los programas siguen una línea de presentación ígual, que puede calificarse como aceptable. Su principal defecto es la poca flexibilidad en el manejo de datos, bastante mejorada en el de mediciones y presupuestos, y la nula gestión de diferentes obras, para poder recalcular o archivar algún dato. También hay que añadir pequeñas picias por falta de un depurado serio. Si no se puede desproteger el programa y acceder al basic para solucionarlas, mal asunto.

Fernando Martinez, Pamplona, 14/4/89

ED:

Puedes utilizar los discos con el QL si tienes un interface de disco e, incluso en formato MS-DOS.

También puedes conectar ambos ordenadores por la puerta serie.

Un emulador de QL para los PCs es totalmente posible por Hardware (imposible por software). Pero quitanto la Card-Futura que nunca fue lanzada debido a la quiebra de SANDY, no conozco ninguna noticia.

Hay rumores de un emulador de QL para el Commodore Amiga en Alemania.

ALGUNAS CURIOSIDADES.

Hace algunos días me escribió un compañero de fatigas ante el teclado desde Salamanca. Su nombre es Jean Alain Rodríguez y creo que está en contacto con otras personas de CUQ; espero que acabará colobarando en este asunto dentro de poco.

Entre otras cosas, Alain me hacía algunas preguntas sobre cuestiones que me parecen de interés general para otras muchas personas, por lo que me he permitido contestar a ellas a través de las páginas de nuestro boletín.

Las preguntas eran las siguientes:

1- ¿Qué es un emulador?

2- ¿Qué es un fractal?

Como podeis ver, la primera cuestión se refiere a un tema plenamente informático, mientras que la segunda entra más de lleno en el campo de las matemáticas, aunque por diversas razones aparece con frecuencia en las revistas sobre informática y existen varios programas que anuncian a bombo y platillo su capacidad para dibujar a gran velocidad figuras fractales, sin que mucha gente sepa a ciencia cierta qué es éso que hacen con tanta rapidez. Pero vayamos por partes y comencemos por el principio:

¿QUE ES UN EMULADOR?

Ultimamente se ha hablado con frecuencia de emuladores en el mundillo del QL; primero emuladores de CP/M y ahora de emuladores de MS-DOS. También existe el caso de un emulador QL para el Atari ST. Bien, entonces ¿Qué es todo este jaleo de ordenadores y sistemas operativos?

El proceso de emulación se reduce básicamente a hacer que un ordenador se crea que es otro ordenador diferente y actúe como lo haría el ordenador emulado: de esta forma es posible conseguir que un ordenador corra programas escritos para otro diferente. Esta es la razón del estallido de emuladores en el mercado del QL: un ordenador que ya no se fabrica tiene unas posibilidades de desarrollo en cuanto a software muy limitadas, sobre todo en un mercado dominado completamente por IBM, ya que las grandes firmas de desarrollo de programas prefieren escribir para un sector de millones de usuarios potenciales y no limitarse a un mercado de unos pocos miles. Así pues, si se puede lograr que el QL corra programas escritos para otros ordenadores de mayor difusión, se habrá prolongado su existencia tanto como la existencia de éstos últimos.

A todo ésto, conviene aclarar que existen dos formas de enfrentarse al proceso de emulación: la primera consiste en emular directamente el funcionamiento de un microprocesador y hacer correr sobre él el sistema operativo original; es el sistema elegido por Digital, tanto en el SUCCESS (que es en realidad un emulador del Z80 que corre un sistema CP/M) como en el THE SOLUTION (que es en realidad un emulador de PC, con su microprocesador y su BIOS o sistema básico de entrada/salida, que es capaz de hacer correr un MS-DOS sobre él). La otra forma consiste en emular directamente el funcionamiento del procesador con el sistema operativo ya instalado, caso del CP/Mulator.

Hasta aquí la teoría, ahora vamos a la realidad. En primer lugar existen dos limitaciones importantes dentro del cuento de hadas de la emulación: para lograr que un procesador interprete las instrucciones que fueron codificadas en el código adecuado para otro diferente es necesario proceder a la traducción una por una de todas las instrucciones según va corriendo el programa: es decir cualquier programa funciona como si fuera interpretado con la consiguiente pérdida de velocidad. Hay que imaginar el proceso como si el emulador fuese tomando cada instrucción del programa a correr, buscase en un gran diccionario la traducción de esa instrucción al set de instrucciones del procesador real que va a ejecutar el programa y entregase dicha traducción para su ejecución. Ahora bien, cualquiera que haya realizado alguna vez una traducción de un idioma a otro sabe que no siempre es posible encontrar en un idioma una palabra que signifique exactamente lo mismo que en el otro, teniendo que recurrir a toda una frase para traducir una palabra única en principio. En el proceso de emulación ésto ocurre a veces y con mayor frecuencia cuanto mayor sea la diferencia entre las filosofías de funcionamiento de ambos procesadores, haciendo aún más lenta la ejecución de un programa que corre sobre un emulador.

En el caso concreto del emulador de MS-DOS para el QL hay que añadir otro factor de reducción de la velocidad: el procesador de Intel emulado es un procesador de 16 bits; por otra parte el 68008 de Motorola que hay dentro de

cada uno de nuestros QLs es internamente un 16-32 bits, pero con un bus de datos de 8 bits, lo que hace que cualquier operación de lectura escritura requiera el doble de tiempo al tener que realizarse en dos pasos consecutivos.

La otra limitación importante a la que me refería es la siguiente: una vez que se ha logrado emular el funcionamiento del microprocesador y problemas de velocidad aparte, se podrá hacer correr casi todo aquello que se haya escrito para acceder a las entradas y salidas de forma ortodoxa (a través del BIOS en un PC), pero no todo aquello que pretenda acceder a las entradas y salidas de forma directa, buscando un hardware que en realidad no existe más que en la "fantasía" del ordenador. Hay que aclarar que este tipo de software no son sólo los juegos a que se refiere la propaganda de Digital Precision, sino también una buena cantidad de software "serio" como copiadores, utilidades de manejo de entrada-salida o programas tan interesantes como Minix, una versión reducida del sistema Unix.

En todo caso y dejando de lado todas las limitaciones anteriores, un emulador, digamos del PC/MS-DOS, puede permitirnos tomar contacto con el estupendo software escrito para los compatibles PC, pero nos aleja de algo mucho más valioso: nuestro sistema operativo QDOS, que es todo un ejemplo de diseño efectivo y compacto.

José Carlos de Prada

¿QUE ES UN FRACTAL?

Creo que hasta aquí ha quedado claro que no entiendo mucho de emuladores: a partir de aquí quedará claro que entiendo mucho menos de matemáticas y por lo tanto de fractales. En todo caso ésto no es la enciclopedia Británica, ni una sala de conferencias de la universidad (en las que, dicho sea de paso se dicen enormes tonterías), por lo que me voy a permitir aportar mi granito de arena al saber (o la ignorancia) general.

Un fractal es una figura geométrica formada por "picos" o puntos de inflexión con una serie de particularidades desde el punto de vista matemático (en las que no voy a entrar por pura ignorancia) y desde el punto de vista estético. Por medio de ecuaciones algebraicas es posible determinar tanto líneas como superficies fractales y por medio de oportunos algoritmos es posible que los ordenadores se dediquen a la tediosa labor de dibujarlos. El resultado puede ser de interés para quien quiera investigar en el comportamiento matemático de estas figuras o para quien simplemente quiera obtener unos dibujos llamativos en su pantalla; además he oido hablar de un juego (escrito para PCs) en el que todos los paisajes que aparecen se forman por medio de superficies fractales. Para el QL es posible encontrar algunos programas capaces de dibujar estas figuras: algunos de ellos utilizan el nombre de Mandelbrot, un investigador de IBM que se dedicó al estudio de algunas de estas curvas.

José Carlos de Prada, Madrid 20-4-89

CP/M

Un saludo a todos.(26-02-1989)

Acabo de recibir el boletin CUQ5 y aunque quizas ya no llege a tiempo mi colaboración para el siguiente número 6,es igual,lo poneis en el próximo.

Os cuento a continuación una historia para ver si alguno de los lectores me hecha una mano y entre todos conseguimos que yo sea capaz de leer un disco de CP/M que me han dejado y que aunque no tengo excesivo interés por los programas que van en su interior,ya es casi una cuestion de honor(lease cabezoneria)el conseguirlo.

En primer lugar os diré que poseo dos emuladores de CP/M el Qlsoft y el de Digital Prec.(SUCCESS).En el primero de ellos hay un fichero llamado RDCPM que sirve para definir o analizar previamente el formato en que está grabado el disco.

Como yo tengo además de la disquetera de 3.5" otra de 3" que me trabaja como Flp2_ uso directamente el disco de 3" formateado y copiado con su propio equipo.Por cierto no lo he dicho antes;el disco a leer es de un PCW8256 de Amstrad.

Analizandolo con el RDCPM me dá los siguientes parámetros:

```
Density..................D
Sector length............512
First sector.............1
Sector/track.............9
```

No. of sides.............1
Use double stepping......no
No. of tracks............30

Después me hace la siguiente pregunta:

Sector skew? (0...5)

Y aqui empiezan los problemas.Cualquier valor entre 0 y 5 que le introduzca solo consigo que me llene la pantalla de 1(unos) de una manera sesgada y sin fin,entrando en una dinamica continua de la que no se salir salvo reseteando el equipo.

He probado también con el SUCCESS pero cuando uso el programa ANALYSE,(no me acepta ninguno de los dos formatos de Amstrad que trae de origen),y aunque le voy cambiando alternativamente diferentes valores posibles para la creación del nuevo formato,después no consigo leerlo.

Tambien he probado con la opcion de lectura de CP/M que hay dentro de la de lectura de discos Alien en el Super Media Manager sin resultado.

Así mismo tampoco lo he sabido hacer con el CPMCOPY,y digo sabido porque me escama que con tantos medios no lo haya conseguido ya.Debe de ser que el operario de dichos programas no da más de sí.

A ver si alguno de vosotros me dais un empujón.

Adjunto a esta carta acompaño un documento Quill que se llama LIB_doc que es la versión en castellano de las instrucciones de QLIBERATOR hecha por mi gran amigo Angel Gil Amores, de Córdoba y que con su permiso os la envio para su difusión si no hay trabas legales.Este compañero,tambien ex-QLAVE,ha relizado unas cuantas versiones castellanas de otros tantos programas en inglés y que es una ayuda inmensa para los que como yo solo dominamos el inglés si es bajito y se deja.

Como no sabemos,tanto él como yo,si con ésto se transgrede alguna norma legal espero para mandar las otras,vuestra indicación.

El enviarlas yo es porque hasta ahora yo he hecho de puente entre él y CUQ y ha recibido a mi través los números anteriores.

Con esto os dejo hasta el próximo mes.Un saludo para todos y ánimo que esto marcha.

Antonio Rodriguez Hernandez.
Aptdo.2107
30000.MURCIA

(EDITOR: Ese disco debe estar corrupto, porque Nacho Enrique no tuvo ninguna dificultad con las definiciones del SUCCESS.

Esta carta ha tenido un retraso ejemplar por motivos de correos y porque la primera entrega llegó corrupta)

---

## NOTICIAS

---

NOTA: Las noticias destinadas a esta sección deben citar la fuente de información de la que se han obtenido.

### 5 MINUTOS EN EXPOELECTRONICA'89

No sé si os acordais (solamente los antiguos socios de QLave) de aquellos 5 minutos de 1987. En aquellos dias os conté que aquello era un intento de vender todos los equipos obsoletos lo más rápido posible (CPC, MSX,...), y que los equipos de 32 bits o basados en un MC 68000 brillaban por su ausencia. Pues os lo creais o no, en solamente un año las cosas han cambiado muchisimo.

Por primera vez, he visto un Apple Macintosh (nunca he visto uno tan cerca) y por si fuera poco, también he visto al increible MAC II (algo antiguillo comparado con el IIx) con monitor monocromo en acción.

Habia Atari ST, Amiga, impresoras matriciales y laser. No habia ni una sola máquina de 8 bits ( ni siquiera el Z88). Los pocos PC-compatibles estaban basados en su mayoria en el Intel 80386 (caballo grande ande o no ande) que es la última moda en PC.

Como podreis observar, los nuevos equipos que dominan el mercado son todos de 32 bits -los 68000 pequeños son de 16/32 bits o 8/32 bits, según se mire- (aunque los PCs de 16 bits aún nos dan la lata gracias a sus precios tan ridiculos). Los discos más usados son los de 3.5". Y yo por mi parte apuesto por los 68000 aunque pierda.

S. Merino, Fuengirola, 20/4/89.

ATARI PC FOLIO Y ATARI ST PORTATIL

Poca información se conoce a cerca del nuevo Atari PC Folio (la poca que existe solamente sirve para confundir al público).

Sus medidas son 18 cm de largo x 10 cm de ancho x 2,7 cm de alto.

Usa pilas normales AA estandar que dependiendo del uso que se haga de la máquina pueden durar de 6 a 8 semanas (o un dia supongo, porque ese debe de ser el tiempo máximo en reposo).

Usa como sistema de almacenamiento unas tarjetas de memoria del tamaño de una tarjeta de crédito que proporcionan 128 K de datos o programas, las cuales se comportan como discos. Atari proporciona opcionalmente una unidad de lectura de estas tarjetas de forma que puedan ser conectadas directamente a cualquier PC. Esto último me suena a quemado, porque me parece que él PC FOLIO el único interface que posee, si es que posee alguno (en el articulo de ATARI User no dicen nada), es el Centronics para impresoras (hay un programa configurador de impresoras).

RAM: 128 K ampliables a 640 K (supongo que externamente, porque poco sitio debe sobrar dentro para meter 512 K)

ROM: 250 K

Microprocesador: Intel 8088 a 4,9 MHz (esto último puede atraer o desorientar al público haciendole creer que es compatible PC o algo parecido, porque un 8088 y un Z80 no se diferencian en mucho e, incluso en el Z88 se ofrece 3 Megas de RAM).

Software en ROM: Hoja de Cálculo (compatible Lotus 1-2-3), Editor de textos, libro de direcciones y Dietario.

Teclado: 63 teclas (si el del Z88 me parece algo justillo, imaginaos este que es del tamaño de una de mi manos. En mi caso particular, para usarlo tengo teclear con un boligrafo).

Pantalla: Tecnologia supertwist, de 11 cm x 4 cm con una resolución de 240x64 (si alguien decia que la del Z88 le parecia pequeña, ¿Qué dirá de esta?).

La fecha de lanzamiento es para Junio 89, pero la versión actual solamente dispone de los lenguajes Inglés, Frances y Alemán. Si es esa la versión para nuestro pais, podemos despedirnos de la "Ñ".

Según mi opinión, me parece uno más de esos ordenadores de bolsillo que venden en los bazares. Si no tiene RS-232 y hay que comprarlo opcional, poco futuro le veo. Y si es indispensable tener un PC para poder usar el lector de tarjetas opcional y tiene éxito, será un producto orientado en exclusiva al mundo PC. E igual que el Z88 no ha hecho disminuir las ventas del Psion Organiser, porque está orientado a un uso en particular, no creo que pueda interferir las ventas del Z88 y el Psion Organiser. Aunque yo apuesto por el Z88 que además de ser portátil, es un ordenador personal como los de verdad y posee una buena cantidad de periféricos (para procesar textos y comunicarse con otros ordenadores es para lo que mejor sirve).

Cuando Atari ha anunciado su portátil STACEY poco futuro le debe estar viendo a su PC Folio. Es un ST portátil con pantalla LCD monocroma con una resolución de 64Ox400 pixels. Disco 3.5" DS/DD 720K, y opcionalmente segunda unidad y disco duro 20 Megas. Se le ha adaptado un "trackball" en el teclado en sustitución al ratón (no pensado para portatiles). La fecha prevista de su lanzamiento son los meses de Junio y Julio 89.

Yo os aconsejaria que cuando compreis un nuevo ordenador, os lo penseis muy bien, porque correis el riesgo de convertiros en coleccionistas de ordenadores (yo ya tengo una colección que no deseo desprenderme), o simplemente consumidores de usar y tirar (comprar el último modelo todos los años y vender el viejo por lo que te den).

Se podria decir que tengo solamente tres modelos en uso : el Spectrum para jugar, el QL para uso profesional y el Z88 para usarlo en cualquier lugar.

S. Merino, Fuengirola, 7/5/1989.

## COMENTARIOS SOBRE HARDWARE

### SEGUIMIENTO DE CINTAS DE MICRODRIVE

Las cintas de microdrive tienen una conocida tendencia a perder eficacia con el tiempo. El abrasivo roce de la cabeza de lectura/escritura las va deteriorando a cada pasada. Esto es inevitable en cualquier dispositivo de cinta o disco magnético en que la cabeza esté en contacto con la superficie magnética (cintas de audio, video, diskettes...), pero en los microdrives el efecto es más patente debido tanto a la gran velocidad de la cinta, como a las características de este, por mucho que se le haya criticado, genial invento de Sinclair (¡cuando pienso en la sencillez del dispositivo, me asombro de que siquiera funcione!). La cinta, dispuesta en un solo bucle sin-fin, es bastante sensible a las agresiones mecánicas: golpes, caidas, manipulaciones poco cuidadosas, polvo, destensamientos... Así, no es nada raro el encontrarse con que, un buen día, aquel modesto cartucho de microdrive en el que, tan confiadamente, conservávamos nuestras extensas memorias, nos dice lastimosamente "Medio incorrecto...", que viene a significar algo como "Adiós, mundo cruel..." en el lenguaje de los wafertapes (nombre correcto de las cintas magnéticas sin-fin).

Así pues, no estaría de más el llevar un control o seguimiento del rendimiento de las cintas, lo cual nos ayudaría a detectar con antelación las que empiezan a no ser fiables. Teniendo diskettes, apenas se hace uso de las cintas, pero, en cualquier caso, el seguimiento de las mísmas siempre es útil para el excepcional caso en que haya que echar mano de alguna. Para el usuario que no disponga de diskettes, es evidente que un tipo de control como el que voy a proponer le será de mucha utilidad.

El control de las cintas consiste en anotar los resultados en sectores que obtenemos con los formateos. Para que esto sirva de algo, es necesario que cada cinta tenga un código de referencia con el cual identificarla. Será necesario además anotar la fecha del formateo, así como el drive en el que fue efectuado éste. Veamos algunos ejemplos absolutamente reales (junto al código de la cinta aparece, entre paréntesis, su origen y fecha de adquisición; la columna "I1" indica los Kbytes obtenidos al formatear la cinta con la Interface 1, en un Spectrum; la columna "QL XZmdv" es para el execepcional caso de emplear un ZX microdrive unido al QL):

Ejemplo de cinta que se mantiene más o menos igual, después de un año:

CINTA: M04 (venía con QL 10/87)

| Fecha | QL mdv1 | QL mdv2 | I1 | QL ZXmdv |
|---|---|---|---|---|
| 010488 | 216/222 | | | |
| | | 213/220 | | |
| | 216/222 | | | |
| | | 214/220 | | |
| | 216/222 | | | |
| | | 214/220 | | |
| 290389 | 215/221 | | | |

Ejemplo de cinta que está para el desguace:

CINTA: M18 (Valente 10/87)

| Fecha | QL mdv1 | QL mdv2 | I1 | QL ZXmdv |
|---|---|---|---|---|
| 010488 | 224/227 | | | |
| | | 219/222 | | |
| | 226/232 | | | |
| | | 219/222 | | |
| | 215/226 | | | |
| | | 220/223 | | |
| 170588 | 219/224 | | | |
| 090888 | 219/224 | | | |
| 240389 | error14 | | | |
| | error14 | | | |
| | | 217/227 | | |
| | | 208/228 | | |
| | | error14 | | |
| | | error14 | | |
| | | error14 | | |

| Fecha | QL mdv1 | QL mdv2 | I1 | QL ZXmdv |
|---|---|---|---|---|
| | | error14 | | |
| | error14 | | | |
| | | 224/227 | | |
| 290389 | error14 | | | |
| | error14 | | | |
| | error14 | | | |
| | | error14 | | |
| | | error14 | | |
| | | error14 | | |
| 300389 | | | 89 | |
| | error14 | | | |
| | | | 85 | |
| | | | 63 | |
| | | | 47 | |
| | | | 47 | |
| | | | 42 | |
| | error14 | | | |
| | | error14 | | |
| | | | 88 | |

Dos ejemplos de cintas "fatales", aunque parece que al Spectrum le sirven:

CINTA: M19 (Valente 10/87)

| Fecha | QL mdv1 | QL mdv2 | I1 | QL ZXmdv |
|---|---|---|---|---|
| 010488 | error14 | | | |
| | | error14 | | |
| | 200/227 | | | |
| | | 208/225 | | |
| | 201/227 | | | |
| | | 201/225 | | |
| 220389 | | error14 | | |
| 290389 | error14 | | | |
| | | error14 | | |
| 300389 | | | 85 | |
| | | | 85 | |
| | | | 84 | |
| | | | 82 | |
| | | | 88 | |

CINTA: M20 (Valente 10/87)

| Fecha | QL mdv1 | QL mdv2 | I1 | QL ZXmdv |
|---|---|---|---|---|
| 010488 | 219/225 | | | |
| | | 218/224 | | |
| | 208/226 | | | |
| | | 214/227 | | |
| | 208/224 | | | |
| | | 210/225 | | |
| 120588 | | | | 205/207 |
| 090888 | error14 | | | |
| | | error14 | | |
| 220389 | 214/225 | | | |
| | error14 | | | |
| | error14 | | | |
| | error14 | | | |
| | | error14 | | |
| | | error14 | | |
| 290389 | | error14 | | |
| | error14 | | | |
| | error14 | | | |
| | | error14 | | |
| 300389 | | | 89 | |
| | | | 89 | |
| | | | 90 | |

Un ejemplo típico de cinta que se ha ido deteriorando poco a poco:

CINTA: M21 (Valente 10/87)

| Fecha | QL mdv1 | QL mdv2 | I1 | QL ZXmdv |
|---|---|---|---|---|
| 010488 | 219/222 | | | |

```
                216/219
        221/224
                217/220
        221/224
                216/219
 090888 216/221
        216/220
        216/220
        218/224
 290389 213/217
```

Como es fácil deducir, "error14" hace referencia a "fallo en inicialización". En resumen, el conservar en un texto todos estos datos, y renovarlos al formatear alguna cinta, pondrá a nuestra disposición una útil visión de la "vida activa" de nuestras cintas de microdrive, con lo que con un solo vistazo sabremos dejar a un lado las cintas que empiecen a dar síntomas de "agotamiento", disminuyendo así el riesgo de desagradables sorpresas futuras.

Marcos Cruz, Madrid, 31/3/1989

----------------------------------------
-------------------------------

LENGUAJE SuperBASIC

-----------------------------------------------------------------------

¡ UN CUQ CON DIBUJOS !

La idea de un CUQ con dibujos no es mia, sino de Marcos Cruz. Pero emocionado por el proyecto, también yo voy a aportar ideas.

Aunque el proyecto de Marcos consiste en mezclar texto con volcados de pantalla uniendolos mediante THE EDITOR, mi idea consiste en hacerlo directamente en un fichero. Para ello necesitamos usar un lenguaje de programación, y como este articulo es para la sección de Superbasic, ya os podeis imaginar cuál será el usado (el Superbasic es algo lentillo, para mi hubiese sido igual de sencillo haber escrito todas las rutinas de este articulo en Assembler o Superforth. Pero eso será otra historia).

Lo primero que debo decir, es que se supone que estamos trabajando en un QL con Trump Card (afortunadamente hay 9 o más lectores con Trump Card).

El fichero de pantalla se crea en un fichero mediante la instrucción SDP_DEV y SDP_SET 1,1,0,0 para la impresora BMC o Epson tipo MX80 (esto se puede hacer también en Superforth sin usar el assembler).

En el Toolkit II hay un comando que practicamente nos hubiese solucionado todo el problema, pero no está preparado para los ficheros de volcados de pantalla (vamos que no funciona ni a tiros). Esa instrucción es SPL.

```
SPL #4 TO #5
```

Hace un volcado del fichero abierto en el canal #4 por el canal #5. Con esta instrucción podemos unir ficheros de texto quedando preparados para enviar a la impresora, pero para los ficheros de volcados de pantalla no sirve (siempre nos da un linea de caracteres por canal #1).

```
100 OPEN_NEW #5,ram1_cuq
110 OPEN_IN #6,ram1_dump1
115 PRINT #5,CHR$(27);"l";CHR$(20)
120 REPeat bucle
130 IF EOF(#6): EXIT bucle
140 PRINT #5, INKEY$(#6);
150 END REPeat bucle
155 PRINT #5,CHR$(27);"l";CHR$(0)
160 CLOSE #5 : CLOSE #6
```

Este programa en Basic consiste en ir cogiendo un sólo carácter de un canal para enviarlo al otro canal. El proceso es lentisimo en Superbasic e incluso en versión compilada tarda un buen rato (el tiempo se mide en minutos. 1 minuto y

45 segundos para la versión Qliberator).
También he probado llenando matriz de cadena y es muy rápido, pero la función EOF no detecta el fin de un fichero.
Los códigos que envia PRINT #5 son para centrar el dibujo que vamos a imprimir por la impresora.
No obstante, aunque esta sección no es la correcta, he escrito la versión anterior en Assembler para comprobar la velocidad.

```
          moveq.l   #-1,d1
          moveq.l   #2,d3
          lea       canal_cuq,a0
          moveq     #1,d0                 io.open
          trap      #2
          lea       salida,a3
          move.l    a0,(a3)
          moveq.l   #0,d3
          lea       canal_dump,a0
          moveq     #1,d0                 io.open
          trap      #2
          lea       entrada,a3
          move.l    a0,(a3)
          moveq.w   #3,d2
          moveq.w   #-1,d3
          move.l    (salida),a0
          lea       margen20,a1
          moveq     #7,d0                 io.sstrg
          trap      #3
bucle
          move.l    (entrada),a0
          moveq     #1,d0                 io.fbyte
          trap      #3
          cmp.w     #-10,d0
          beq       fin_fichero
          move.l    (salida),a0
          moveq     #5,d0                 io.sbyte
          trap      #3
          bra.s     bucle
fin_fichero
          moveq.w   #3,d2
          lea       margen0,a1
          move.l    (salida),a0
          moveq     #7,d0                 io.sstrg
          trap      #3
          move.l    (entrada),a0
          moveq     #2,d0                 io.close
          trap      #2
          move.l    (salida),a0
          moveq     #2,d0                 io.close
          trap      #2
* Mensaje de final de trabajo
          move.w    #11,d2
          move.w    #-1,d3
          move.l    #$0,a0
          lea       terminado,a1
          moveq     #7,d0                 io.sstrg
          trap      #3
          move.l    #-1,d1
          move.l    #0,d3
          moveq     #5,d0                 mt.frjob
          trap      #1
canal_cuq
          dc.w      8
          dc.b      'ram1_cuq'
canal_dump
          dc.w      10
          dc.b      'ram1_dump1'
margen20
          dc.b      27,108,20
margen0
          dc.b      27,108,0
```

```
entrada
          ds.l      1
salida
          ds.l      1
          dc.w      11
terminado dc.b      'Fin trabajo'
          end
```

La versión es idéntica a la original, pero con la diferencia de que enviamos al canal #0 el mensaje "Fin Trabajo", y el tiempo usado es de 20 segundos (el programa ocupa 196 bytes). Por lo demás, no tiene ningún misterio y supongo que ayudará a alguien a comprobar que programar en Assembler bajo QDOS es fácil.

Un detalle muy importante a destacar es que no podemos usar el comando TRA 1 cuando estamos imprimiendo gráficos. Lo que significa que debemos instalar el QUILL o THE EDITOR para que traduzca los caracteres Españoles cuando haga un volcado de impresora a un fichero. Con los demás editores habria que escribir un pequeño programa que hiciese una nueva copia con las traducciones (nada del otro mundo, os lo aseguro).

De todas formas, todo esto sigue siendo versiones experimentales que darán lugar a la final. También habrá que probar en assembler leyendo muchos bytes de una sola vez, pues asi se gana muchisimo tiempo.

El proyecto solamente tiene un único defecto, todos debemos tener la misma impresora o compatible. Que yo sepa, practicamente la mayoria (el 90%) tenemos una BMC o compatible MX80.

¡ Uno de estos nuevos ficheros podria ser impreso con cualquier ordenador (PC, ATARI ST, Spectrum,.....) si usa la misma impresora !

S. Merino, Fuengirola, 6/4/89.

PROGRAMA PARA HACER "DESCENDER" JUEGOS DE CARACTERES

Esta sencilla rutina permite, a partir de un juego de caracteres para el QL, crear otro en el que los caracteres estén "bajados" un cierto número de pixels. El programa parmite manipular juegos de caracteres con cualquier número de éstos, automáticamente. Está configurado para caracteres de 9 scans de altura (los normales), pero puede ser modificado si es preciso; también puede modificarse el programa para "subir", en lugar de "bajar", los caracteres.

```
100 REMark baja_charsets_bas
110 REMark Programa para hacer descender un número determinado
120 REMark de pixels un juego de caracteres
130 :
140 REMark Marcos, 30/1/1989
150 :
160 TK2_EXT:REMark Toolkit II
170 pixels=2:REMark pixels que hay que bajar los caracteres
180 INPUT "Nombre del juego de caracteres="!nombre$
190 bytes=FLEN(\nombre$)
200 dirección=ALCHP(bytes)
210 LBYTES nombre$,dirección
220 CHAR_USE#1,dirección,0
230 p_carácter=PEEK(dirección)
240 n_caracteres=PEEK(dirección+1)
250 dir_char=dirección+2
260 FOR carácter=0 TO n_caracteres
270   FOR veces=1 TO pixels
280     FOR scan=0 TO 8
290       POKE dir_char+8-scan,PEEK(dir_char+7-scan)
300     END FOR scan
310     POKE dir_char,0
320     AT 1,0
330     FOR muestra=p_carácter TO p_carácter+n_caracteres
340       PRINT CHR$(muestra);
350     END FOR muestra
360   END FOR veces
370   dir_char=dir_char+9
380 END FOR carácter
390 CHAR_USE 0,0
400 INPUT "Nombre el nuevo juego de caracteres="!nombre$
410 SBYTES nombre$,dirección,bytes
```

420 RECHP dirección

Marcos, Madrid, 31/3/1989

PROGRAMA DE TENIS PERFECCIONADO

En el CUQ 7 se incluyó un programa de tenis, como aquellos primeros "ping-pong" que causaron tanto furor en los tiempos "pre-Sinclairianos". La pega es que el programa sólo podía ser utilizado por dos jugadores, es decir, que no se podía jugar contra la máquina.

Unas sencillas y rápidas modificaciones me han permitido convertirlo en "TENIS PLUS", donde no sólo puede jugarse contra la máquina, sino que además puede jugar ella solita contra sí misma. El programa está en el fichero tenis_plus_bas que se incluye con este número de CUQ. La velocidad de ejecución es aceptable, de modo que no añado versión compilada (además, ¿quién no tiene un compilador a estas alturas del siglo?). Compilado, y además con las extensiones aceleradoras de gráficos de Lightning, debe ir de maravilla (no lo he probado aún, pues interpretado funciona a buena velocidad).

Comparando ambos listados, se apreciará la sencillez de las modificaciones efectuadas. Al comienzo, puede elegirse entre 0, 1 o 2 jugadores. A la hora de mover las raquetas, el programa atendererá el estado del teclado o de la pelota, según cada una haya de ser movida por el jugador o el ordenador, respectivamente.

Estas son las líneas y variables añadidas para pedir el número de jugadores:

```
INPUT "Número de jugadores (0-2)",jugadores
jugador_der=0
jugador_izq=0
IF jugadores:jugador_der=1
IF jugadores>1:jugador_izq=1
```

Y estas las que realizan el nuevo control de las raquetas:

```
DEFine PROCedure explorar
IF KEYROW(1)=8:GO TO 3350
IF jugador_der
IF KEYROW(1) = 128 : toca_raqueta_der_abajo
IF KEYROW(1) = 4 : toca_raqueta_der_arriba
ELSE
  IF flag
    IF (y_der+6)>y
      toca_raqueta_der_arriba
    ELSE
      toca_raqueta_der_abajo
    END IF
  END IF
END IF
IF jugador_izq
  IF KEYROW(0) = 1 : toca_raqueta_izq_arriba
  IF KEYROW(0) = 32 : toca_raqueta_izq_abajo
ELSE
  IF flag=0
    IF (y_izq+6)>y
      toca_raqueta_izq_arriba
    ELSE
      toca_raqueta_izq_abajo
    END IF
  END IF
END IF
END DEFine explorar
```

¡Veréis como no es tan fácil darle una paliza! ¡Más bien al contrario!

Marcos Cruz, Madrid, 31/3/1989

-----------------------------------------------------------------------

COMENTARIOS DE PROGRAMAS

-----------------------------------------------------------------------

TRUCOS CON THE EDITOR

The Editor es un potentísimo programa de tratamiento de textos que permite la manipulación de éstos en casi todas las formas imaginables, con la ventaja de que sus comandos pueden a su vez escribirse en ficheros de texto, constituyendo

así pequeños programas o rutinas que pueden ser ejecutadas en cualquier ocasión.

Para dar alguna muestra de la potencia de este programa, mostraré un par de pequeños ejemplos de cosas que, mediante unas sencillas órdenes, pueden hacerse para manipular textos en minutos; tareas que, hechas a mano, llevarían horas de trabajo:

PRIMER EJEMPLO:

Imaginemos que tenemos escrito un texto en dos columnas, por ejemplo un poema y su traducción a otro idioma, uno al lado del otro. Ahora imaginemos que se nos ocurre la feliz idea de cambiar de lado ambos poemas: el de la derecha a la izquierda, y el de la izquierda a la derecha. Con un tratamiento de textos normal, la tarea sería sobrehumana: línea a línea, cortando, definiendo bloques, moviendo, volviendo a unir... ¡pero aquí está The Editor!:

```
rp(39cr;s;p;bs;be;n;bm;39cr;j;n)
```

Para quien no conozca el significado de los comandos de The Editor, ahí va una pequeña explicación:

```
rp   repetir lo que sigue
39cr 39 veces mover el cursor a la derecha
s    partir la línea por la posición del cursor
bs   marcar inicio de bloque
be   marcar final de bloque
n    mover el cursor al inicio de la línea siguiente
bm   mover el bloque a la posición del cursor
39cr 39 veces mover el cursor a la derecha
j    unir línea del cursor con la siguiente
n    mover el cursor al inicio de la línea siguiente
```

En lugar de "rp", puede ponerse el número de líneas de que conste el texto a alterar, y en lugar de "39" debe ponerse el número menos 1 de la columna en que comiencen los versos de la columna de texto de la derecha.

Así, con sólo esa orden, estando el cursor en la primera línea del texto, The Editor hará todo el trabajo por nosotros.

SEGUNDO EJEMPLO:

Ahora, supongamos que queremos pasar un fichero fuente de ensamblador Z80 proveniente de un ensamblador GENS3 de un Spectrum, al QL, para ensamblarlo con otro ensamblador. El traspaso es sencillo, pues basta tener conectadas ambas máquinas mediante RS232 (o red local, también serviría), hacer un OPEN#3,"b" en la Interface 1 del Spectrum antes de ejecutar el GENS3, hacer un COPY SER2C,FLP1_fichero en el QL, y por último un listado a "impresora" (o sea, al QL) desde el GENS3, del texto en cuestión. Cuando el Spectrum termine de emitir, pulsamos CTRL SPACE en el QL y ¡ya está!

Cuando carguemos el listado Z80 nos encontraremos con que tiene dos pegas: en primer lugar, cada línea tiene un número de línea, lo cual no nos vale, y en segundo lugar los parámetros de cada comando están siempre en la misma columna, lo cual significa que existe un número variable de espacios entre ambas partes de las órdenes. Es decir, necesitamos convertir cada línea de algo como:

```
9450 LOOP   LD    A,B
```

a algo como:

```
LOOP   LD A,B
```

¿Manos a la obra? Lo primero que se nos pasa por la cabeza es emplear la teoría de Abundio, experimentada en el Zentro Zuperió d'inbetigazióne zientifíca de Lepe: armarse de paciencia y corregir línea a línea a mano. Pero ¡tachán! ¡The Editor asoma en el horizonte!:

```
jl;200(6cr;s;p;d;7cr;s;a//;p;pr;d;2p;7cr;j;n)
```

Veamos qué hace eso:

```
jl   justificar a la izquierda
200  repetir 200 veces lo que sigue
6cr  6 veces mover el cursor a la derecha
s    partir la línea por la posición del cursor
p    mover el cursor a la línea anterior
d    borrar la línea del cursor
7cr  7 veces mover el cursor a la derecha
s    dividir la línea por la posición del cursor
a//  añadir una línea en blanco después de la del cursor
p    mover el cursor a la línea anterior
pr   reformatear el párrafo
d    borrar la línea del cursor
2p   dos veces mover el cursor a la línea anterior
7cr  7 veces mover el cursor a la derecha
j    unir la línea del cursor con la siguiente
n    mover el cursor a la línea siguiente
```

Ante nuestros asombrados ojos, The Editor comenzará a partir, borrar,

formatear y volver a unir líneas y líneas, mientras nos podemos ir a dar un paseo. "200" es el número de líneas a modificar; "7" es el ancho del campo de etiquetas, que puede alterarse según sea preciso (aunque es lo más habitual que sea 7), y "6" debe ser siempre el ancho del campo de etiquetas menos 1. El realizar estas operaciones lleva su tiempo, y , si el número de líneas es respetable, podemos ir a prepararnos un té con pastas y todo, pero mejor eso que perder todo un día borrando espacios de sobra delante de la pantalla.

En resumen, The Editor es la releche. Lo mismo que hemos hecho con The Editor podíamos haberlo hecho mediante un programa en SuperBASIC u otro lenguaje... ¿pero, por qué nos íbamos a meter en esa historia, si el mismo editor de textos puede hacer la tarea?

Los ejemplos mostrados corresponden a la versión 1.098 de The Editor. La última versión creo que permite muchas más sofisticaciones, como definir bloques menores de una línea y cosas parecidas, lo cual habría simplificado y acelerado las operaciones.

Lo más importante para diseñar rutinas para The Editor es fijarse dónde queda el cursor después de ejecutar cada orden. Para ello, nada mejor que hacer unas pruebas, introduciendo las órdenes una a una y observando los resultados, antes que arriesgarnos a hacer un galimatías con los textos. Con un poco de lógica, casi cualquier cosa puede hacerse con este estupendo programa.

La (mil veces maldita) pega de The Editor es que, al estar escrito en SuperBASIC y compilado con Supercharge, plantea problemillas con el Toolkit II y con Turbo Toolkit. Si estuviera escrito en 68000, y así no necesitase ese fichero de extensiones que lleva, sería perfecto.

Marcos Cruz, Madrid, 30/3/1989

LLAMAMIENTO

A quien conozca bien alguno de estos programas, le agradecería que lo comentase muy a fondo (todo: in/utilidad, in/eficacia, in/compatibilidades, des/ventajas, pegas, fallos...):

QLOAD, DISCOVER, SUPER MEDIA MANAGER, QRAM, TASKMASTER, SPY, THE EDITOR (última versión)

Las cuestiones de fondo son:

- Velocidad de QLOAD (para evitar la eterna carga del SuperBASIC)
- SPY versus THE EDITOR (para escribir programas y textos)
- QRAM versus TASKMASTER (para todo)
- DISCOVER versus SUPER MEDIA MANAGER (para copiar QDOS<->MS/DOS)

¿Lo entendéis? Pues eso.

Gracias.

Marcos cruz, Madrid, 1/4/1989

PROGRAMA : DOMINATION
DISTRIBUIDOR : PDQL
PRECIO : 9 LIBRAS

En primer lugar hay que decir que es un juego de estrategia (he descubierto la rueda).

El objetivo del juego es dominar el mundo militarmente y económicamente.

El juego es de 2 a 4 jugadores (uno o varios puede ser el QL).

Los paises son representados por cuadrados.

Cada pais produce un determinado producto y todos producen armas.

El juego consiste en comprar, vender y hacer la guerra al vecino. Y como en la pelicula de los inmortales, "solamente puede quedar uno".

Según PDQL, el juego es adictivo (es algo parecido al juego de tablero RISK. En la mili llegué a jugar a ese juego hasta con mi Oficial de Seguridad ¡Adivinad mi divisa! Si os sirve de pista, mi relevo usa el chaleco anti-balas que solicité, pero aún no me he enterado de qué pasó con mi tercer par de botas), pero no he tenido tiempo para comprobarlo, pues aqui hace falta mucha imaginación.

La portada de este mes se la he dedicado a este juego para que os hagais una idea de lo vais a ver (deja mucho que desear si se le compara con juegos parecidos del SPECTRUM de hace 5 años).

Según mi opnión, los juegos no es el punto fuerte del QL, pero no podemos decir lo mismo de los programas de utilidades. Aunque eso no quiere decir que el QL no sirve para juegos, simplemente que los usuarios QL no están precisamente interesados en juegos.

S. MERINO, Fuengirola, 13/4/1989

PROGRAMA : QL CADETTE

DISTRIBUIDOR : STRONG COMPUTER SYSTEMS
PRECIO : 29 LIBRAS

QL CADETTE es un programa para dibujar planos en 2 dimensiones. Pero su principal atractivo es poder usar una buena colección de PLOTTERs.

El verdadero propietario de la copia que estoy comentando posee el SILVER REED 4 COLOUR PEN PLOTTER, pero no tiene tiempo para poder comentarnos todo lo que posee (aunque como ya estais viendo, no es un problema grave).

Al cargar el BOOT se nos da a elegir entre cargar CADETTE (es el programa CAD en si) o PLOT (es el programa para volcados en impresora Epson y dibujos en PLOTTER).

El programa CADETTE es muy fácil de usar, quizás no haga falta ni leerse las instrucciones (bastante claras).

Todo es hecho a base de menus y la pantalla de dibujo está formada por coordenadas cartesianas.

Cualquier cosa dibujada o alterada es almacenada en una Database (seguramente para su uso posterior por el PLOTTER).

Se puede cambiar la escala y el tamaño de hoja (A4 o A3).

Se pueden dibujar : Arcos, cajas, circulos, elipses, lineas, poligonos regulares, texto y triangulos.

Naturalmente la colección de datos debe ser almacenada en un disco o MDV para su posterior uso, o para imprimir con impresora o Plotter con el programa PLOT.

El Plotter Driver soporta los Plotters Hewlwt-Packard compact usando el lenguaje HPGL), Roland DXY y Penman. Y las impresoras Epson.

QL Cadette es compatible Qram y Speedscreen (y es Turbo compilado). Corre en máquinas 128K.

Me ha parecido un buen paquete de dibujo lineal.

S. MERINO, Fuengirola, 13/4/1989

CALCULO DE VIGAS TRIANGULADAS.

consta de los ficheros VIGAS1, VIGAS2, VIGAS3, BOOT, y los de fichero de materiales (todos los datos de las vigas)

El programa viene en dos mdvs, pero si se desprotege, cabe en uno solo, aunque la carga es mucho mas lenta al ser en basic. De todas formas, con estos programas, si no los tenemos en basic accesible es mejor tirarlos a la basura porque ya he dicho que están sin depurar.

En VIGAS1 se carga la definición geométrica de la viga, que ha de tener las barras justas, no admitiendo ni mas ni menos que las necesarias para triangularla. Si se quiere meter alguna barra más por razones estéticas, o se supone que no trabaja, o se mete con truco. (P.E. en vez de una barra del nudo A al B, meter tres del nudo A1 al A2 y al B, separando A1 y A2 un milímetro.)

Las barras se dan metiendo las coordenadas de los nudos, y luego definiendo las barras por los nudos de sus extremos. La entrada es lenta pero segura. Ahora bien, la corrección es muy mala, ya que solo se puede acceder inmediatamente despues de meter los datos, y vuelven a salir estos secuencialmente uno a uno.

Al final dibuja la viga e imprime los resultados, pero no crea un fichero con el nombre de la estructura y sus datos, sino que el fichero es genérico.

También se definen auí las constantes generales, como tipode acero, coeficientes de seguridad etc...

Los defectos que le he pillado son:

La máxima coordenada Y no puede ser mayor que la X al meter los datos. (Cosa bastante lógica por cierto)

En las coordenadas Y solo coge dos decimales en vez de tres.

Si la viga es muy complicada, el dibujo no lo hace con los nombres de las barras, sólo con el de los nudos. Aún así, puede que no se vean mas que los números de los nudos.

Sería muy importante el volcado a impresora del dibujo, con los nombres de las barras y los nudos. Si no ha de hacerse a mano.

En VIGAS2 el programa es muy parecido al anterior, pero entra las cárgas, las hipotesis y combinaciones de las mismas.

Aquí ya se puede liar todo. Confunde hipótesis con combinaciones. Por ejemplo:

Las hipótesis son cuatro: Viento, Nieve, Personas, Máquina.

Las combinaciones son dos:Que pegue el Viento, que haya Nieve, que esté lleno de Personas y que estén las máquinas, o como segundo caso, que estén sólo las máquinas.

Pues bien, al meter las combinaciones sería H1+H2+H3+H4 y H3+H4. El programa no me acepta mas que la segunda, ya que cree que como hay dos combinaciones no

puede haber cuatro hipótesis.

Los últimos datos a meter son los tipos de perfiles de las barras. Hay para elegir entre los clásicos, y no permite diseñar un perfil propio o combinar varios perfiles que no estén ya metidos. Hay IPN, UPN, dobleC, T, Doble T y angulares.

Al acabar la entrada de datos, la mecánica de corrección es la misma que ántes. Desde lugo, no podemos modificar datos de la definición geométrica, porque esa parte del programa sólo guarda los datos metidos en un fichero mdv2_, y el programa está borrado.

Otras picias que he pillado son:

Al salir los datos por impresora, no salen todas las cárgas. Por ejemplo:

Viento tiene 20 cargas, una para cada nudo. Nieve tiene 15, porque los nudos verticales no almacenan nieve. Pues solo imprime las cargas de los 15 nudos cargados en los dos casos.

He comprobado que los cálculos son correctos.

Para imprimir en DIN A4 hay que modificar en los programas la línea 175 y 114. Donde pone if nli<=59, poner 54.

Hay que tener cuidado con el sentido de las fuerzas y ver el ángulo de incidencia para determinar el signo de las mismas.

VIGAS3 es el programa que calcula los perfiles. La salida de datos por impresora se hace a la vez que va calculando las barras.

Los datos pueden obtenerse en versión reducida y nudo a nudo, saliendo los siguientes datos:

Ya de VIGAS1 y VIGAS2 se imprime el título general, el de las hojas con numeración, el valor de las coordenadas de los nudos, definición de las hipótesis de cargas y valores de las fuerzas, cargas a sumar en combinación de hipótesis, perfiles a utilizar en cada barra, esfuerzos en las barras para cada hipótesis y nudo a nudo, marcando si es esfuerzo tracción o compresión, y los grados de inclinación de la barra. Un resumen de cargas en barras por hipótesis. Combinaciones de hipótesis de cargas. Cargas más desfaborables. Esfuerzos de compresión en barras con carga máxima de tracción. Y por último el cálculo de perfiles en barras, donde da: Tipo de angulares, la carga, la longitud de la barra, la sección en cm2. el rádio mínimo de giro la esbeltez y el coeficiente de pandeo si está comprimida, y la tensión del acero.

Sin haber analizado los cálculos internos del programa, creo que tiene alguna fálta. Las cargas se introducen en los nudos, pero la carga de las vigas superiores producida por las correas creo que no la considera. En determinados casos se puede despreciar, pero en otros es imprescindible su cálculo. Yo he tenido que recalcular estas barras a mano. No es mucho trabajo una vez que está todo ya calculado.

En resumen:

Al programa le falta mucha interelación con el usuario, de tal forma que el técnico pueda diseñar la estructura con relación a los resultados que va viendo. Si el resultado final le sale de mucho peso, o con pocas barras etc.. que pueda cambiar algun dato sin tener que repetir de arriba a abajo todo el programa.

Como anticipo al comentario de Mediciones Y presupuestos, tengo que decir que he encontrado la forma para que se imprima la opción "mediciones y presupuestos conjuntos" en la impresora. Es una pequeña variable que nos ha hecho la vida imposible al no poder trabajar con mediciones y presupuestos conjuntamente. Una picia de las más gordas del autor.

En el fichero ZY11 en la línea 1539 poner F$=" " en vez del valor que le da. Todo solucionado. (de momento)

Y esto es todo por ahora.

FERNANDO MARTINEZ, Pamplona, 14/4/89

--------------------------------------------------------------------------------

TRADUCCIONES DE MANUALES

--------------------------------------------------------------------------------

PRIMERA ENTREGA (CONSTA DE DOS ENTREGAS). HAY UNA VERSION ORIGINAL.

SUPERTOOLKIT II

VERSION 2.06

MANUAL DEL USUARIO

Versiones del TOOLKIT

Las nuevas versiones del Toolkit difieren de la versión 2.00 en los siguientes aspectos:

Se revisa el BREAK (CTRL ESPACIO) durante el WCOPY y el WREN incluso si se ha solicitado A(LL) (TODOS).

Se ha añadido PRINT_USING y FEXT$. PRINT_USING es más comprensible que en la forma dada en el manual redactado.

El servidor de ficheros de la red local ha sido ampliado para incluir dispositivos en serie (impresoras), además de mensajerías de QL a QL. Se ha cambiado el comando NFS_USE con el fín de darle mayor flexibilidad. En particular, varios usuarios pueden ahora compartir un disco de datos usando QUILL. El protocolo del servidor de ficheros en red local de la versión 2.00 no es compatible con las nuevas versiones.

El parche para la ROM MG ha sido suprimido para hacer sitio a lo anterior (Nota: este parche se incluye todavía en las versiones en microdrive).

Aviso Obligatorio

QL, QL NET, QDOS, SuperBASIC y SINCLAIR son marcas registradas de Sinclair Research Limited, Reino Unido.

Copyright Tony Tebby, 1985

Reservados todos los derechos. Ninguna parte de este logicial o de su documentación puede ser reproducido de ninguna manera. La copia, alquiler, préstamo, venta y reventa no autorizadas se encuentran prohibidas.

En ninguna circunstancia pueden ni Care Electronics ni Qjump ser responsabilizados de cualquier daño o pérdida directo, indirecto o consecuencial que pueda ocurrir por cualquier error, defecto o fallo de la versión ROM del SuperToolkit II.

Care Electronics y Qjump mantienen una política de continuo desarrollo y mejora de sus productos, y mantendrán siempre informados a los usuarios legítimos y registrados de este producto sobre los cambios y mejoras a las cuales se encuentre sujeto.

Manual del usuario, edición inglesa escrita por Tony Tebby, Qjump, Reino Unido. Traducido por E. de Jesús.

INDICE DE MATERIAS

Prefacio

El QL Toolkit original fué producido con precipitación para proveer de facilidades útiles que debieran haber estado en el QL desde el principio. Desde su aparición, he estado sometido a contínua presión para modificar ciertas facilidades y para aumentar el número de posibilidades ofrecidas.

El QL Toolkit II es, por tanto, una versión revisada (hasta el extremo de estar casi completamente reescrita) y muy alargada del QL Toolkit original. Las antiguas facilidades trabajan ahora más rápidamente y son más compactas, de manera que hay sitio en el cartucho de ROM para más de 100 comandos.

El heho de que el QL Toolkit II haya visto la luz del día ha sido debido a incitaciones de diferentes procedencias. Alguna gente contactó conmigo para quejarse de que habían sido incapaces de conseguir el QL Toolkit original, lo que eventualmente me convenció de que había un mercado para una segunda versión. Chass Dillon hizo críticas repetidas y de gran calibre (y con justificación) a las facilidades originales, lo que puso la base para la modificación de algunas de las antiguas rutinas. El aporte práctico de Ed Bradley ha sido invaluable al poner el producto en el mercado, y Cambridge Systems Technology me permitió usar uno de sus sistemas de disco Winchester para probar el servidor de red local.

El QL Toolit II no podría tampoco haber sido completado sin el indisminuible entímulo de Hellmuth Stuven de QSOFT, Dinamarca, cuya indominable confianza en el mérito técnico de este producto me ha mantenido en alerta.

Mis agradecimientos a todos vosotros

Tony Tebby

El Toolkit II de QJUMP para el QL

El Toolkit II de QJUMP para el QL es una versión alargada y mejorada del QL Toolkit original. Esta versión ha sido ampliamente reescrita con el fin de proveer más facilidades y para hacer las existentes en el QL y en el Toolkit más poderosas.

Ya que muchas de estas mejoras son para corregir defectos del las ROMs suministradas con el QL, hubiera sido mejor suministrar otras para reemplazar a las de Sinclair. Dada la actitud no cooperativa de Sinclair Research Limited hacia una tal mejora, este Toolkit II se suministra de la mejor forma posible.

---------------------------------------------------

1 Introducción

El Toolkit II intenta poner un gran número de facilidades en una forma
SuperTOOLKIT II CARE ELECTRONICS - QJUMP -3-
consistente. Un pequeño preámbulo es bueno para explicar algunos de los principios.

Este manual usa la siguiente convención sencilla para describir los comandos y las llamadas de función:

```
LETRAS MAYUSCULAS     para las partes tecleadas tal cual
letras subrayadas     se usan como descripción
letras minúsculas     se usan como ejemplos
Así
VIEW nombre           es una descripción
VIEW fred             es un ejemplo
```

1.1 Comandos Procedimientos Funciones

Las extensiones al SuperBasic aparecen como comandos, procedimientos, y funciones. La distinción entre un comando y un procedimiento es muy débil y los dos nombres pueden ser usados intercambiablemente: el comando es lo que un usuario teclea, el procedimiento es lo que hace el trabajo.

En algunos casos, un comando invoca un procedimiento que a su vez selecciona e inicializa una tarea (por ejemplo SPL comienza el spooler residente). Una función es algo que tiene un valor y el nombre de una función no puede ser usado

como un comando: el valor puede ser impreso (PRINT), usado en una expresión o asignado a una variable.

1.2 Y/N/A/Q?

Y/N/A/Q? es una concisa, aunque quizás inicialmente confusa, pregunta que el QL puede hacer de vez en cuando al usuario. Este debe pulsar una de las teclas Y (para YES-SI), N (para NO), A (para ALL-TODOS), o Q (para QUIT-ABANDONAR).

Lo que ocurra al presionar una de esas teclas depende de lo que esté intentando usted hacer en ese momento.

Existe una forma corta en la que solo se pregunta Y (para SI) y N (para NO).

Cualquier contestación diferente a la pregunta Y/N/A/Q? (o Y or N?) es descartada. Pulsar BREAK (CTRL + espacio) o ESC tiene el mismo efecto que pulsar "Q" (o"N").

1.3 Sobreescritura

En algunos casos, se da un comando para crear un nuevo fichero con el nombre de uno que ya existe. En general, en lugar de un mensaje de error, aparecerá una pregunta pidiendo permiso para sobreescribir ("OK to overwite... Y or N?").

Hay dos (deliberadas) excepciones a esta regla:

OPEN_NEW devolverá un error, mientras que los procedimientos COPY_O, SAVE_O, SBYTES_O, SEXEC_O y el spooler sobreescribirán en los ficheros de destino.

1.4 #canal

Todas las entradas y salidas del SuperBasic se realizan a través de "canales". Algunos de estos canales son implícitos y nunca son vistos (por ejemplo, el comando "SAVE SER" abre un canal a SER, lista el programa en el canal, y cierra el canal). Otros se identifican por un número de canal que es un entero pequeño y positivo precedido por un # (por ejemplo, #2).

Algunos comandos permiten o requieren que sea especificado un canal para entradas y salidas, el cual debe se un número de canal del SuperBasic.

#0 es el canal para los comandos (en la parte inferior de la pantalla)

#1 es el canal normal para las salidas

#2 es el canal para el listado de programas

Otros canales (por ejemplo, para comunicación con un fichero) pueden ser abiertos usando los comandos OPEN del SuperBasic (ver sección 10).

Para comandos interactivos el canal de omisión es el #0, para la mayoría de los comandos restantes es el #1, para LIST y ED el canal de omisión es el #2, mientras que para comandos de acceso a ficheros la omisión es el #3.

Para algunos de los comandos es posible especificar un canal implícito, el cual tiene la forma de "\" seguido de un nombre de fichero o de dispositivo. Su efecto es el de abrir un canal implícito al fichero o al dispositivo, hacer la operación requerida y cerrar de nuevo el canal.

Por ejemplo,

DIR lista el directorio actual en #1

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -4-

DIR #2 lista el directorio actual en #2

DIR\dlist lista el directorio actual en el fichero dlist

Este último ejemplo debe ser distinguido de

DIR dlist lista en #1 las entradas de directorio que comiencen por dlist

1.5 Nombres de Fichero y de Dispositivo

En general, es posible especificar los nombres de ficheros o de dispositivos como un nombre normal en SuperBasic o como una cadena. La sintaxis de los nombres en SuperBasic limita los carácteres usados en el nombre a letras, dígitos y subrayado. No existe tal limitación en los nombres usados en una cadena. En un QL estándard, los nombres de fichero deben ser dados en su forma completa, pero usando el Toolkit II, la parte de directorio del nombre puede ser omitida, dando únicamente el nombre del fichero. Por ejemplo,

OPEN #3, fred abre el fichero "fred" en el actual directorio

Ello produce un problema: el intérprete de SuperBasic tiene la característica poco afortunada de tratar de evaluar todos los parámetros de un comando como expresiones; en el ejemplo anterior, "fred" será probablemente una variable no definida que no dará problemas. Sin embargo, el comando:

OPEN #3, list
dará un error de "expresión errónea" ya que no es posible para "LIST", que es un comando, tener un valor. Hay una manera de rodear el problema: evitar nombres de fichero iguales a comandos(procedimientos), funciones o palabras clave del SuperBasic (por ejemplo, FOR, END, IF, etc), o poner el nombre entre comillas, como una cadena:
OPEN #3,'list' o OPEN #3,"list"

1.6 CTRL F5
La pulsación de CTRL F5 (mantener pulsada la tecla de CONTROL y pulsar F5) se usa para congelar la pantalla del QL. Algunos comandos del Toolkit II observan su salida a pantalla, y cuando ésta está llena provocan internamente un CTRL F5 para detener la salida hasta que el usuario pulse una tecla (la mejor tecla a pulsar será habitualmente F5).

----------------------------------------------------

2 Contenido del Toolkit II

El SuperBasic en el QL se usa como lenguaje de programación y como lenguaje de comandos. Se proveen extensiones para mejorar las facilidades del SuperBasic en ambas áreas, así como herramientas para el desarrollo de programas.
La lista siguiente da una visión sumaria de cada comando o función. Los parámetros tienen a menudo valores de omisión para simplificar el uso de los procedimientos.

2.1 Facilidades de Desarrollo
Sección 3 Editando Ficheros
El toolkit II provee de un editor y de un comando para ver el contenido de los ficheros de texto. ED es un editor basado en ventana para editar programas en SuperBasic. VIEW es un comando para examinar ficheros basados en líneas (por ejemplo, ficheros fuente en ensamblador).
Comandos
ED #canal,número de línea edita programa en SuperBasic
VIEW #canal,nombre ver el contenido de un fichero

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -5-
2.2 Lenguaje de Comandos
Las facilidades del lenguaje de comandos del Toolkit II intentan abrir el QL al potencial del sistema operativo QDOS. Muchos de ellos son comandos "directos": actúan no más ser tecleados. Ello no significa que no deban ser empleados en programas, pero al hacer esto debe tenerse cierto cuidado.

Sección 4 Control de Directorios
El QDOS tiene un sistema de ficheros con una estructura de directorios en árbol. El Toolkit II aporta una selección comprensible de comandos para controlar el acceso a los directorios dentro de este árbol.
Comandos
DATA_USE nombre selecciona el directorio de omisión para ficheros de datos
PROG_USE nombre selecciona el directorio de omisión para programas ejecutables
DEST_USE nombre selecciona el directorio de omisión de destino (COPY, WCOPY)
SPL_USE nombre selecciona el dispositivo de omisión de destino (SPL)
DDOWN nombre baja a un subdirectorio
DUP sube a través del árbol
DNEXT nombre mueve a otro directorio en el mismo nivel
DLIST #canal lista los valores de omisión
Funciones

DATAD$ devuelve
actual directorio para datos
PROGD$ devuelve
actual directorio para programas
DESTD$ devuelve
actual directorio destino

Sección 5 Mantenimiento de Ficheros

Todos los comandos de mantenimiento del sistema de ficheros usan los directorios de omisión (normalmente de "datos"). Algunos de los comandos son interactivos y, por tanto, no apropiados para su uso en programas en SuperBasic: son los que se encuentran marcados con un asterisco en esta lista. En esos casos, hay comandos más simples que pueden ser usados en los programas. Dependiendo del comando, el nombre dado puede ser un nombre genérico ("wildcard") que se refiera a más de un fichero. Con la excepción de DIR (una versión extendida del comando DIR estándard del QL), todos estos comandos empiezan por "W".

Comandos

DIR #canal,nombre estadística
del dispositivo y
lista de ficheros
WDIR #canal,nombre lista de
ficheros
STAT #canal,nombre estadística
del dispositivo
WSTAT #canal,nombre lista de
ficheros y su
estadística
DELETE #nombre borra un
fichero
*WDEL #canal,nombre borra
ficheros

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -6-

COPY nombre TO nombre copia un
fichero
COPY_O nombre TO nombre copia un
fichero(sobreescribe)
COPY_N nombre TO nombre copia un
fichero(sin cabecera)
COPY_H nombre TO nombre copia un
fichero(con cabecera)
*WCOPY #canal,nombre TO nombre copia
ficheros
SPL nombre TO nombre vuelca un
fichero
SPLF nombre TO nombre vuelca un
fichero, con <FF> al
final
RENAME nombre TO nombre renombra un
fichero
*WREN #canal,nombre TO nombre renombra
ficheros

Sección 6 Programas en SuperBasic

El Toolkit II redefine y extiende las operaciones de salvaguarda y carga de ficheros del QL. Todos los comandos usan directorios de omisión. Adicionalmente, los comandos de control de ejecución han sido modificados para atender a las funciones de manipulación de errores de las ROMs "JS" y "MG".

Comandos

DO nombre ejecuta
comandos en fichero
LOAD nombre carga un
programa en SuperBasic
LRUN nombre carga y
ejecuta un programa en SuperBasic
MERGE nombre une un
programa en SuperBasic
MRUN nombre une y
ejecuta un programa en SuperBasic
SAVE nombre,rango salva un

programa en SuperBasic
SAVE_O nombre,rango idem pero
sobreescribe el
fichero, si existe
RUN número de línea comienza un
programa en SuperBasic
STOP para un programa en SuperBasic
NEW reinicializa el SuperBasic
CLEAR borra las
variables de SuperBasic

Sección 7 Cargar y Salvar

Las operaciones de carga y salvaguarda han sido modificadas para que usen los directorios de defecto

Comandos
LRESPR nombre carga un
fichero en el área de
procedimientos residentes y lo llama (CALL)
LBYTES nombre,dirección carga un
fichero en la
dirección especificada de memoria
CALL dirección,parámetros llama al
código máquina con
parámetros
SBYTES nombre,dirección,tamaño salva un
área de memoria

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -7-

SBYTES_O nombre,dirección,tamaño como SBYTES
pero, si existe
sobreescribe el fichero
SEXEC nombre,dirección,tamaño,datos salva un
área de memoria en
la forma de fichero ejecutable
SEXEX_O nombre,dirección,tamaño,datos como SBYTES
pero, si
existe,sobreescribe el fichero

Sección 8 Ejecución de Programas

La ejecución de programas es, Anne Boleyn debería estar aliviada de saverlo, lo contrario de (ex)terminación. En el QL Toolkit los comandos EXEC y EXEC_W del QL estándard fueron sustituidos por EX y EW. El Toolkit II redefine EXEC y EXEC_W para que sean lo mismo que EX y EW. ET es sólo para depuradores de programas (Nota del Traductor: la palabra inglesa para "depuradores de programas" significa literalmente deschinchadores, por lo que Tony Tebby añade aquí: sin intención de ofender).

Comandos
EXEC/EX especificaciones del programas carga y
activa uno o más
ficheros ejecutables
EXEC_W/EW especificaciones del programa
ET especificaciones del programa

Sección 9 Control de Tareas

Las facilidades de multitarea del QDOS se hacen accesibles mediante los comandos y funciones para control de tareas del Toolkit II.(id=identificación).

Comandos
JOBS #canal lista las
tareas actuales
RJOB id o nombre,código de error suprime una
tarea
SPJOB id o nombre,prioridad selecciona
la prioridad
de una tarea
AJOB id o nombre,prioridad activa una
tarea

Funciones
PJOB(id o nombre) devuelve
prioridad de la tarea
OJOB(id o nombre) devuelve el
propietario de tarea

JOB$(id o nombre) devuelve el nombre de la tarea
NXJOB(id o nombre) devuelve la siguiente tarea en el árbol

2.3 Programando en SuperBasic

El Toolkit II tiene extensiones al SuperBasic para ayudar en la escritura de programas más potentes y flexibles. Las mayores mejoras se encuentran en la manipulación de ficheros y en el formato de entradas y salidas.

Sección 10 Abrir y Cerrar

Los comandos estándard del QL para apertura de canales son redefinidos por el Toolkit II para usar los directorios de omisión. Además, el Toolkit II aporta un grupo de funciones para la apertura de ficheros bien especificando un número de canal (como en los comandos estándard del QL), o bien encontrando ellos mismos, y devolviendo, un número de canal vacante. Las funciones permiten también la intercepción y proceso por los programas en SuperBasic de los errores del sistema de ficheros.

Comandos

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -8-

OPEN #canal,nombre abre un fichero para lectura/escritura
OPEN_IN #canal,nombre abre un fichero sólo para lectura
OPEN_NEW #canal,nombre abre un nuevo fichero
OPEN_OVER #canal,nombre abre un nuevo fichero y, si existe, lo sobreescribe
OPEN_DIR #canal,nombre abre un directorio
CLOSE #canales cierra canales

Funciones

FTEST (nombre) revisa el estado de un fichero
FOPEN(#canal,nombre) abre un fichero para lectura/escritura
FOP_IN(#canal,nombre) abre un fichero sólo para lectura
FOP_NEW(#canal,nombre) abre un nuevo fichero
FOP_OVER(#canal,nombre) abre un nuevo fichero y, si existe, lo sobreescribe
FOP_DIR(#canal,nombre) abre un directorio

Sección 11 Información de Ficheros

El Toolkit II tiene un grupo de funciones para leer la información de la cabecera de un fichero.

Comandos

FLEN(#canal) devuelve la longitud del fichero
FTYP(#canal) devuelve le tipo de fichero
FDAT(#canal) devuelve el espacio de datos del fichero
FXTRA(#canal) devuelve la información extra del fichero
FNAME$(#canal) devuelve el nombre del fichero
FUPDT(#canal) devuelve la fecha de actualización del fichero

Sección 12 Acceso Directo a los Ficheros

El Toolkit II tiene un grupo de comandos para transferrir datos a y de cualquier parte de un fichero. Los comandos leen o escriben ellos mismos datos "crudos", tanto en forma de bytes individuales como en el formato interno del SuperBasic (entero, coma flotante o cadena).

Comandos

BGET #canal\posición,items toma bytes de un fichero

BPUT #canal\posición,items pone bytes en un fichero

GET #canal\posición,items toma datos de un fichero
en formato interno

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -9-

PUT #canal\posición,items pone datos en un fichero
en formato interno

TRUNCATE #canal\posición trunca un fichero

FLUSH #canal nivela los amortiguadores de un fichero

Funciones

FPOS(#canal) devuelve la posición en el fichero

Sección 13 Conversiones de Formato

El Toolkit II aporta un número de facilidades para entradas y salidas formateadas. Se incluyen conversiones de números binarios y hexagesimales, así como formato fijo de decimales.(nd=número de decimales)

Comandos

PRINT_USING #canal,formato salida en formato fijo

Funciones

FDEC$(valor,campo,nd) formato fijo decimal

IDEC$(valor,campo,nd) ídem

CDEC$(valor,campo,nd) ídem

FEXP$(valor,campo,nd) formato fijo exponencial

HEX$(valor,número de bits) conversión a hexagesimal

BIN$(valor,número de bits) conversión a binario

HEX(cadena hexagesimal) hexagesimal a valor

BIN(cadena binaria) binario a valor

Sección 14 Control de la Pantalla

El Toolkit II tiene comandos para activar y desactivar el cursor, para seleccionar la fuente y tamaño de caracteres y para restaurar las ventanas al estado en que se encuentran cuando se conecta el aparato.(d=dirección)

Comandos

CURSEN #canal activa el cursor

CURDIS #canal desactiva el cursor

CHAR_USE #canal,d1,d2 selecciona o reinicializa la fuente de caracteres

CHAR_INC #canal,inc x,inc y selecciona los incrementos x e y de los caracteres

WMON modo reinicializa a monitor

WTV modo reinicializa a televisión

Sección 15 Administración de la Memoria

El Toolkit II tiene un grupo de comandos y funciones que aportan facilidades para la administracipon de la memoria del área de la "pila común" del QL.
Funciones
FREE_MEM devuelve la cantidad de memoria libre
ALCHP(número de bytes) reserva espacio en la pila común
(devuelve la dirección de la base del espacio)
SuperTOOLKIT II CARE ELECTRONICS - QJUMP -10-
Comandos
RECHP dirección base devuelve espacio a la pila común
CLCHP limpia todas las reservas en la pila común
DEL_DEFB borra de la pila común los bloques
de definiciones de ficheros

Sección 16 Parámetros de los Procedimientos
Cuatro son las funciones que tiene el Toolkit II para mejorar la manipulación de los parámetros de los procedimientos (y funciones). Usándolos es posible determinar el tipo (entero,coma flotante o cadena) y el uso (valor simple o matriz) de un parámetro de llamada así como su "nombre".
PARTYP(nombre) devuelve el tipo de parámetro
PARUSE(nombre) devuelve el uso del parámetro
PARNAM$(número de parámetro) devuelve el nombre del parámerto
PARSTR$(nombre,número de parámetro) si el parámetro "nombre" es una cadena, devuelve su valor,
si no, encuentra la cadena

Sección 17 Manipulación de Errores
Se aportan las siguientes facilidades para el proceso de errores en las versiones JS y MG del SuperBasic.
ERR_DF verdadera si ha ocurrido el error de disco lleno
REPORT #canal,número de error informa de un error
CONTINUE número de línea continua en la línea dada
RETRY número de línea
vuelve a la línea dada

Sección 18 Mantenimiento del Tiempo
Con el Toolkit II se proveen dos relojes, un reloj digital configurable, y un reloj-alarma.
CLOCK #canal,formato reloj con formato variable
ALARM horas,minutos reloj-alarma

Sección 19 Extras
EXTRAS lista las facilidades extras conectadas
al SuperBasic
TK2_EXT refuerza las definiciones del Toolkit II
para los comandos y funciones comunes

2.4 Extensiones a los Dispositivos
Además del las extensiones al intérprete de SuperBasic, el Toolkit II tiene otras que afectan a los controladores de consola, de microdrive y de red local.

Sección 20 Controlador de Consola
El Toolkit II posibilita la recuperación de la última linea tecleada en el canal de comandos (#0) así como la asignación de una cadena de caracteres a la pulsación de <ALT> y un caracter.

También, para las versiones de ROM MG, corrige los problemas que hay con los comandos para el dibujo de puntos (POINT), líneas (LINE) y arcos (ARC) cortos en

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -11-

dichas ROMs.

Comandos

<ALT><ENTER> Pulsación
que recupera la
última línea tecleada

ALTKEY caracter,cadena asigna una
cadena a una
pulsación de <ALT> y un caracter

Sección 21 Controlador de Microdrive

El Toolkit II mejora el controlador de microdrive para proveerle de apertura (OPEN) de ficheros con sobreescritura, además de truncado (TRUNCATE) y renombrado (RENAME) de ficheros. Estas facilidades son soportadas tanto a nivel de QDOS (Traps #2 y #3) como de SuperBasic. La operación FLUSH se encuentra reespecificada para actualizar la cabecera del fichero además de nivelar los amortiguadores.

Sección 22 Controlador de Red Local

El controlador de Red Local ha sido mejorado para proveer de una primitiva forma de comunicación de emisión, así como de un programa "servidor de ficheros" compacto el cual permite a varios QLes compartir un sistema de disco o una impresora.

Comandos

FSERVE invoca al
"servidor de ficheros"

NFS_USE nombre,nombres de red local selecciona el nombre del
servidor de ficheros en red local

Nombres de dispositivos

Nnúmero de estación_dispositivo el nombre
de un dispositivo
remoto de entradas y salidas
(p. ej., N2_FLP1_ es el disco 1 de la
estación de red local 2)

-------------------------------------------------

3. Editando Ficheros

3.1 ED - Editor de SuperBasic

ED es un pequeño editor para programas en SuperBasic cargados en el QL. Si sus facilidades parecen mas bien cortas y limitadas, recuérdese que el mayor requerimiento del diseño de ED es su pequeño tamaño para dejar sitio a otras facilidades.

ED es invocado tecleando:

ED
o ED número de línea
o ED #número de canal
o ED #número de canal, número de línea

Si no se da el número de línea, se listará la primera parte del programa; de lo contrario, el listado en la ventana comenzará en o después del número de línea dado. Si no se da número de canal, el listado aparecerá en la ventana normal del SuperBASIC para edición de programas (#2). Si se da una ventana y ésta no es del tipo CONsola, se devolverá el error "parámetro incorrecto". El editor usará los colores actuales de papel y tinta para el listado, usando tinta blanca y papel negro (o al revés, si el papel es ya negro o azul) para resaltar. Por favor, evita utilizar la ventana #10 para ED.

El editor hace uso completo de su ventana. Dentro de ella, intenta mostrar líneas enteras. Si una línea es demasiado larga para entrar en el ancho de la ventana, se la continúa en la siguiente fila de la ventana: estas filas extras dan claridad a la presentación. Sin embargo, para facilidad de uso, debe usarse una ventana lo más ancha posible.

ED no debe ser llamado desde el interior de un programa en SuperBASIC.

La tecla ESC se usa para volver al modo de comandos del SuperBASIC.

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -12-

Después de invocar ED, las teclas de cursor mueven el cursor para seleccionar la línea que va a ser editada. Además las teclas ARRIBA y ABAJO pueden ser utilizadas con la ALT (pulsando la tecla ALT y, manteniéndola

pulsada, presionando la tecla ARRIBA o ABAJO) para enrrollar la ventana, manteniendo el cursor en su lugar; las teclas ARRIBA y ABAJO también pueden usarse con la de MAYUSCULAS para enrrollar el programa de página en página.

El editor tiene dos modos de operación: inserción y sustitución (para cambiarlos pulse MAYUS F4). No hay diferencia entre ambos modos cuando se añaden o borran caracteres al final de una línea. Sin embargo, en medio de una línea, en el modo de inserción, la parte derecha de la línea se mueve hacia la derecha cuando se añade una caracter o hacia la izquierda cuando se borra. En el modo de sustitución, no se mueve ninguna parte de la línea. Los espacios situados al final de la línea se eliminan automáticamente.

Para insertar una línea en cualquier parte del programa, pulsa ENTER; se abrirá un espacio por debajo de la línea actual (aquella sobre la que se encuentra el cursor) y se generará un nuevo número de línea. Si no hay espacio entre la línea actual y la que le sigue (por ejemplo, el caso en que la línea actual es la 100 y la siguiente la 101) se ignora la pulsación de ENTER. Si la diferencia entre el número de línea actual y el siguiente es mayor de de 20, el nuevo número de línea será el actual más 10; si la diferencia es menor, el nuevo número será uno intermedio entre ambos.

Si se hace un cambio en una línea, ésta será resaltada: esto indica que la línea ha sido extraída del programa. El editor no repondrá la línea hasta que se pulse ENTER, se mueva el cursor fuera de la línea o se enrrolle la ventana, momento en el cual, si la línea es aceptable para el SuperBASIC, será reescrita sin resaltar. En el caso en que hubiera algún error de sintáxis, se enviaría el error "línea incorrecta" a la ventana #0, permaneciendo la línea resaltada.

Mientras una línea permanezca resaltada, se puede emplear ESC para recuperar la copia original de la línea, ignorándose todos los cambios que se le hayan hecho.

Si se cambia un número de línea, la línea antigua seguirá en su lugar y se insertará una nueva línea en el lugar correcto del programa. De esta forma, se pueden copiar líneas individuales de una parte del programa a otra.

Una línea será borrada del programa cuando se borren todos sus caracteres visibles, o se deje sólo el número de línea. Una forma más sencilla de borrar la línea es pulsando CTRL/ALT/IZQUIERDA. La longitud de las líneas está limitada a unos 32766 caracteres. Cualquier intento de editar líneas más largas, causará efectos indeseables. Si al cambiarla, se aumenta la longitud de una línea, puede haber una ligera pausa mientras el SuperBASIC mueve su espacio de trabajo.

3.2 Sumario de las Operaciones de Edición

El uso general de las teclas está de acuerdo con la sección de Conceptos de la Guía del Usuario del QL y con el dado en los programas de aplicaciones.

TAB tabulación hacia la derecha (8 columnas)

MAYUS TAB tabulación hacia la izquierda (8 columnas)

ENTER aceptar o crear nueva línea

ESC escapar - anular cambios o volver al SuperBASIC

ARRIBA mover el cursor una línea arriba

ABAJO mover el cursor una línea abajo

ALT ARRIBA enrrollar la pantalla una línea hacia arriba

ALT ABAJO enrrollar la pantalla una línea hacia abajo

MAYUS ARRIBA enrrollar la pantalla una página arriba

MAYUS ABAJO enrrollar la pantalla una página abajo

IZQUIERDA mover el cursor un caracter a la izquierda



DERECHA mover el cursor un caracter a la derecha

CTRL IZQUIERDA borrar caracter a la izquierda del cursor

CTRL DERECHA borrar caracter bajo el cursor

MAYUS F4 cambiar entre modo de inserción y sustitución

3.3. Viendo un Fichero

VIEW es un procedimiento diseñado para examinar un fichero en una ventana de la pantalla de QL. La ventana de omisión es la #1.

View es invocado tecleando

VIEW nombre Ver el fichero "nombre" en la ventana #1

VIEW #canal,nombre Ver fichero "nombre" en el canal dado

VIEW nombre1,nombre2Enviar fichero "nombre1" a "nombre2"

VIEW parte las líneas para poder meterlas en el ancho de la ventana. Cuando la ventana se llena, se genera un CTRL F5. Si el dispositivo de salida no es una consola, las líneas se parten a los 80 caracteres.

4. Control de Directorios

4.1 Estructuras de Directorio

En la terminología del QDOS, un directorio es el lugar donde el sistema espera encontrar un fichero. Un directorio puede ser tan sencillo como el nombre de un dispositivo (por ejemplo, el nombre del Microdrive 2: MDV2_) o mucho más complejo, formando parte de un "árbol de directorio". Por ejemplo, el directorio MDV2_ puede incluir los directorios JOHN_ y OLD_ (nota: todos los nombres de directorio acaban en "_"), y JOHN puede incluir los ficheros DATA1 y TEST.

```
                      MDV2_
       JOHN_                    OLD_
DATA1        TEST
```

Obsérvese que los árboles de directorio crecen hacia abajo. El nombre de fichero completo en el QDOS para DATA1 en el ejemplo anterior es MDV2_JOHN_DATA1.

Una característica inusual de la estructura de directorios del QDOS es la ausencia formal de una "extensión" al nombre de fichero: no son estrictamente necesarias ya que las "extensiones" (como _aba para los ficheros de ABACUS o _asm para los ficheros fuente en ensamblador) son tratadas como ficheros dentro de un directorio.

Lo dicho puede ser ilustrado en el caso de un programa en ensamblador TEST, procesado usando el macro ensamblador y editor de GST. El fichero fuente en ensamblador (TEST_ASM), el listado de salida del ensamblador (TEST_LIST), la salida relocalizable del ensamblador (TEST_REL), el fichero de control del enlazador ("linker") (TEST_LINK), el listado de salida del enlazador (TEST_MAP) y el programa ejecutable producido por el enlazador (TEST_BIN) son todos tratados como ficheros del directorio TEST_.

```
                                                    MDV2_
                                 JOHN_
          TEST_
ASM   LIST   REL   LINK   MAP   BIN
```

El Toolkit II permite seleccionar directorios de omisión. Los valores de omisión se pueden utilizar en todas las operaciones del sistema de ficheros. Un valor de omisión puede ser seleccionado en cualquier nivel de complejidad, dando el punto de partida para encontrar un fichero en la estructura de árbol. Así, en el ejemplo anterior, si el valor de defecto es MDV2_, bastará dar el nombre JOHN_TEST_ASM para encontrar el fichero fuente en ensamblador. Si el valor de omisión es MDV2_JOHN, dicho fichero será encontrado en TEST_ASM. Con el nombre entero MDV2_JOHN_TEST_ASM, se encontrará el fichero independientemente del valor de omisión.

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -14-

4.2 Seleccionando los Valores de Omisión

Inusualmente, las extensiones del Toolkit II al QDOS, posibilitan tres valores de omisión distintos para la estructura de directorio. Ello es debido a que el QDOS es intrínsecamente un sistema operativo multi-dispositivo. Es de esperar que los programas ejecutables estén en un directorio diferente, y probablemente también en un dispositivo diferente, que los ficheros con los datos manipulados.

Además, los procedimientos de copia serán usados más probablemente para copiar de un directorio a otro, o a una impresora o otro dispositivo de salida, que para copiar al mismo directorio.

Hay tres comandos para seleccionar los tres valores de omisión:

DATA_USE nombre directorio selecciona omisión datos

PROG_USE nombre directorio selecciona

omisión programas
DEST_USE nombre directorio selecciona omisión destino

Si el nombre de directorio dado no acaba en "_", el subrayado será añadido al nombre de directorio.

El valor de DATA_USE se utiliza para la mayoría de comandos del sistema de ficheros del Toolkit. PROG_USE se usa sólo para buscar los ficheros de programas en los comandos EX/EXEC. DEST_USE se usa para encontrar el nombre de fichero destino cuando se usan los comandos que copian o renombran ficheros (SPL, COPY, RENAME, etc) con un solo nombre de fichero.

Hay una forma especial de DEST_USE que no añade "_" al nombre dado. Dicha forma define el dispositivo de omisión del "spooler":

SPL_USE nombre de dispositivo

De esta forma se selecciona también el dispositivo de omisión, pero si el nombre del dispositivo no acaba en "_", no se le considera un directorio, de manera que el valor de omisión para el nombre de fichero de destino, se utiliza sin modificación.

Así, en los dos ejemplos siguientes:

```
DEST_USE flp2_old  (el valor de omisión es FLP2_OLD_)
......
SPL fred
o
SPL_USE flp2_old_ (el valor de omisión es FLP2_OLD_)
......
SPL fred
```

el fichero "fred" será volcado a "flp2_old_fred". Sin embargo, si, como en el ejemplo siguiente, se utiliza SPL_USE con un nombre que no acaba en "_" (es decir, un nombre que no sea de tipo directorio):

```
SPL_USE ser     (el valor de omisión es SER)
......
SPL fred
```

el fichero "fred" será volcado a "ser" en lugar de a "ser_fred".

Obsérvese que SPL_USE y DEST_USE definen en mismo valor de omisión (el de destino).

4.3 Navegación a través de los Directorios

Hay tres comandos que permiten moverse a través de un árbol de directorio.

DDOWN nombre desciende (añade "nombre" al valor de omisión)
DUP asciende (elimina el último nivel del directorio)
DNEXT nombre asciende y a continuación desciende por una rama diferente del árbol

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -15-

No es posible ascender más allá del nombre del dispositivo mediante el comando DUP. La longitud del valor de omisión no puede de ninguna manera exceder los 32 caracteres.

Estos comandos operan sobre el directorio de omisión para los datos. Bajo ciertas condicionas, pueden también actuar sobre los otros valores de omisión:

Si el valor de omisión para programas es el mismo que para datos, ambos valores se unen y los comandos anteriores operarán también sobre el valor de omisión para PROG_USE. Si el valor de omisión para la destinación, acaba en "_" (es decir, si es un directorio y no un dispositivo), los comandos anteriores actuarán también sobre dicho valor.

A continuación de da un ejemplo de la aplicación de las reglas anteriores:

| | datos | programas | destino | valores iniciales |
|---|---|---|---|---|
| mdv2_ | mdv1_ | ser | DDOWN john | mdv2_john_ |
| mdv1_ | ser | DNEXT fred | mdv2_fred_ | mdv1_ |
| ser | PROG_USE mdv2_fred | mdv2_fred_ | mdv2_fred_ | ser |
| DNEXT john | | mdv2_john_ | mdv2_john_ | ser |
| DUP | mdv2_ | mdv2_ | ser | DEST_USE mdv1_ |
| mdv2_ | mdv2_ | mdv1_ | DDOWN john | mdv2_john_ |
| mdv2_john_ | | mdv1_john_ | SPL_USE ser1c | |

mdv2_john_ mdv2_john_ ser1c

4.4 Orientándose

Mediante el siguiente comando, que lista los tres valores de omisión, podrás saber en qué nivel del árbol de directorios te encuentras:

DLIST lista valores de omisión para datos,
o DLIST #canal programas y destinaciones
o DLIST \nombre

Cuando no se especifica canal de salida, los valores de omisión se listan en la ventana #1.

Existen tres funciones que permiten encontrar los valores de omisión desde un programa en SuperBASIC:

DATAD$ devuelve la omisión para datos
PROGD$ devuelve la omisión para programas
DESTD$ devuelve la omisión para destinaciones

Estas funciones se usan sin ningún parámetro:

```
IF DATAD$<>PROGD$: PRINT "Directorios Separados"
DEST$ = DESTD$
IF DEST$(LEN(DEST$))="_": PRINT "Destino"!DEST$
```

Tanto en el Sinclair QL Toolkit original, como en este Toolkit, existen facilidades para permitir a los programas ejecutables el encontrar los valores de omisión de los directorios, aunque estas facilidades apenas son empleadas por los programas comerciales para el QL.

Lo solución real para permitir el uso de directorios de omisión al nivel de "traps" del QDOS, sólo puede ser obtenida mediante el uso de accesorios adicionales en la puerta para expansiones o mediante el cambio de las ROMs que contienen el sistema operativo. Sin embargo, la mayoría de los programas comerciales no reconocerían los valores definidos. En el apéndice A se da un ejemplo de como superar este problema.

---------------------------------------------------

5 Mantenimiento de Ficheros

Los procedimientos estándard para el mantenimiento de ficheros (COPY, DELETE y DIR) han sido ampliados a un juego racional en el Toolkit II. Tanto los comandos nuevos como los antiguos, usan los directorios de omisión; además, algunos de los comandos usan unos nombres especiales llamados "wild card" para



referirse a grupos de ficheros con nombres similares.

5.1 Nombres "Wild Card"

Un nombre "Wild card" es un tipo especial de nombre de fichero en el que una parte de dicho nombre puede ser sustituida por cualquier cadena de caracteres. Al diseñar los nombres "Wild card" hemos tenido en cuenta que es conveniente que dichos nombres cumplan las normas sintácticas del SuperBasic, por lo que en ellos no pueden usarse caracteres especiales (por ejemplo, el nombre misficheros_*_asm para indicar todos los ficheros que comienzan por misficheros_ y acaban por _asm, sería tratado por el SuperBasic como una expresión arimética, por lo que trataría de multiplicar misficheros_ por _asm).

Por esta razón, se ha adoptado un esquema más simple: los nombres de ficheros se dividen en secciones (flp2_fred_bin tiene tres secciones) y los nombres "Wild card" pueden tener secciones perdidas (por ejemplo, flp2_old__list tiene una sección perdida). El nombre "Wild Card" refiere a todos los ficheros cuyos nombres tengan secciones que emparejen con las secciones del "Wild Card". El final de un nombre "Wild card" se considera implícitamente como perdido.

Si el nombre "Wild card" no es el nombre completo de ningún fichero, se pone el nombre del directorio de omisión delante del comienzo del nombre.

En los siguientes ejemplos, se supone que FLP2_ es el directorio de omisión:

Nombre
Nombre "Wild card" Ejemplos de ficheros
"Wild card" entero designados por "Wild card"
fred
flp2_fred flp2_fred
flp2_freda_list _fred

flp2__fred flp2_fred
flp2_freda_list
flp2_old_fred
flp2_old_fred_list flp1_old__list flp1_old__list
flp1_old_jo_list
flp1_old_fred_list

5.2 Listado de Directorios

Hay dos formas de listado de directorios: La primera lista solamente los nombres de los ficheros mientras que la segunda lista también su tamaño y la fecha de la última modificación. Todos los comandos usan los nombres "Wild card" y el directorio de omisión para datos. La salida de todos los comandos, por omisión, es el canal #1; sin embargo, puede especificarse otro canal o un dispositivo. Si la salida es a pantalla, el listado se para cuando la ventana está llena.

DIR #canal, nombre estadística del disco y listado de los ficheros
WDIR #canal,nombre listado de los ficheros
WSTAT #canal,nombre listado de los ficheros y de sus estadísticas

En todos los casos, la especificación tanto del canal como del nombre es optativa.

Las formas posibles de, por ejemplo, WDIR son:

WDIR lista el directorio actual en #1
o WDIR #canal lista el directorio actual en #canal
o WDIR \nombre lista el directorio actual en "nombre"
o WDIR nombre lista el directorio "nombre" en #1
o WDIR #canal,nombre lista el directorio "nombre" en #canal

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -17-

o WDIR \nombre1,nombre2 lista el directorio "nombre2" en "nombre1"

Por ejemplo:

WDIR \ser1,_asm lista todos los ficheros "_asm" del directorio actual en SER1
WDIR flp1_ lista todos los ficheros de FLP1_ en #1
WDIR #3 lista todos los ficheros del directorio actual en el canal #3

Se incluye el comando DIR sólo por compatibilidad: antes del listado de los ficheros, aparece la estadística del dispositivo (nombre del medio, número de sectores vacantes/número de sectores buenos).

5.3 Estadística del Dispositivo

Hay un comando que imprime la estadística del dispositivo que acoge a un directorio especificado o al directorio de omisión.

STAT #canal,nombre
o STAT \nombre1,nombre2

Tanto el canal como el nombre son opcionales.

5.4 Borrado de Ficheros

Se ha modificado el procedimiento estándard DELETE de manera que, cuando no se le de un nombre de fichero completo, use los directorios de omisión. No se genera ningún error de "no encontrado". Hay también un comando interactivo que permite el borrado de grupos de ficheros usando nombres "Wild card".

DELETE nombre borra un fichero
WDEL #canal,nombre borra ficheros

Para WDEL tanto el canal como el nombre son opcionales. Por ejemplo:
WDEL borra ficheros del directorio actual
WDEL _list borra ficheros _list del directorio actual
A no ser que se especifique un canal, el procedimiento de borrado "Wild card" usa la ventana de comandos #0 para pedir confirmación para el borrado. Hay cuatro posibles respuestas:
Y(yes-si) borra este fichero
N(no) no borres este fichero
A(all-todos) borra este y todos los ficheros siguientes
Q(quit-abandonar) no borres ni este fichero ni el resto
(Nota: Las versiones reducidas del Toolkit -aquellas que vienen en la ROM del controlador de disco, incorporan otro comando, WDEL_F que funciona como WDEL con la diferencia que el borrado de los ficheros es Forzado, es decir no se pide permiso para ello)

5.5 Copia de Ficheros
Se han cambiado las dos formas del comando COPY para que utilicen los nombres de omisión de ficheros, y además para darles mayor flexibilidad. También se han añadido algunos comandos más.
Los ficheros en el QDOS tienen una cabecera que da información útil sobre el fichero que sigue.
En ciertas circunstancias puede ser útil copiar la cabecera del fichero que sigue. Este es el caso cuando:
a) se copia un fichero de un programa ejecutable de manera que se conserva la información del fichero.
b) se copia un fichero a través de una unión en serie pura de forma que el

logicial de comunicación sabrá de antemano la longitud del fichero.
En otras, sin embargo, no será una buena idea copiar la cabecera. Este es el caso cuando:
c) se copia un fichero de texto a la impresora ya que es probable que contenga códigos de control o caracteres no imprimibles o falsos.
La regla general usada por los procedimientos COPY en el Toolkit II es que la cabecera sólo se copia si contiene alguna información adicional. De esta forma, se atiende a los casos a) y c) anteriores. Se incluye un comando COPY_N por compatibilidad con el mismo comando estándard del QL: este comando nunca copia la cabecera. Además se incluye un nuevo comando, COPY_H, el cual copia siempre la cabecera, y debe usarse en el caso b) anterior (nótese que el comando estándard COPY copia siempre la cabecera). Ni COPY_H ni COPY_N necesitan ser usados para la copia de fichero a fichero.
Una segunda regla usada por el procedimiento COPY (y además por el WREN) es que, si el fichero destino realmente existe, el usuario será requerido para confirmar si la sobregrabación del fichero es aceptable. El procedimiento COPY_O (copia con sobregrabación) y los del "SPooLer" no tienen esta deferencia con el usuario.
Si se utilizan los comandos con dos nombres de fichero, se usará el directorio de omisión para datos en ambos casos. Si, sin embargo, sólo se da un nombre (o ninguno, en el caso de los procedimientos "Wild card"), el fichero de destino se derivará del directorio de omisión para el destino.
a) si la omisión para el destino es un directorio (acabado en "_", seleccionado con DEST_USE), el fichero de destino es el valor de omisión para el destimo, seguido del nombre.
b) si la omisión para el destino es un dispositivo (no acabado en "_", seleccionado con SPL_USE) el destino es el valor de omisión para el destino sin modificación.

5.5.1 Copias de un sólo Fichero
COPY nombre TO nombre copia un fichero
COPY_O nombre TO nombre copia un fichero (sobregrabando)
COPY_N nombre TO nombre copia un fichero (sin cabecera)
COPY_H nombre TO nombre copia un ficehro (con cabecera)

Estos comandos pueden entrarse con uno o dos nombres. Se usa el separador "TO" por claridad, pero puede usarse una coma en su lugar.

Para ilustrar estos comandos de copia, he aquí unos ejemplos en los que se asume que el directorio de datos es MDV2_ y el de destino MDV1_:

COPY fred TO old_fred copia mdv2_fred a mdv2_old_fred

COPY fred, ser copia mdv2_fred a ser

COPY fred copia mdv2_fred a mdv1_fred

SPL_USE fred

COPY fred copia mdv2_fred a ser

5.5.2 Copiando Varios Ficheros

El comando interactivo de copia WCOPY se usa para la copia de todo o parte de un directorio. El comando puede ser dado con nombre "Wild Card" fuente y destino, con un sólo nombre "Wild Card", o, incluso, con ninguno. Este último caso equivale a dar un sólo nombre nulo:

WCOPY y WCOPY "" son lo mismo

Si las siguientes reglas sobre la derivación del nombre fichero destino de la copia, te parecen confusas, usa WCOPY intuitivamente, mirando con cuidado las preguntas que te hace.

Si el destino no es el directorio de omisión para destino, el nombre del fichero de destino verdadero para cada operación de copia se obtiene a partir

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -19-

del nombre de fichero verdadero fuente y del nombre "Wild Card" de destino. Si una sección perdida del nombre "Wild" fuente está emparejada con una sección perdida del nombre "Wild" destino, esta parte del nombre verdadero del fichero fuente será usada como la parte correspondiente del nombre de fichero destino. Si no, el nombre verdadero para el fichero destino se toma del nombre "Wild" de destino. Si hay más secciones en el nombre "Wild" de destino que en el fuente, las secciones extras serán insertadas detrás del nombre del dispositivo y viceversa.

La forma completa del comando es:

WCOPY #canal,nombre TO nombre copia ficheros

El separador TO se usa por claridad y puede ser sustituido por una coma.

Si no se da el canal, las preguntas para confirmación son enviadas al canal de comandos #0 (Nota: #1, para algunas versiones del Toolkit reducido), de manera que en dicho canal (o en el escogido) el usuario será requrido para que pulse:

Y (yes-si) copia este fichero

N (no) no copies este fichero

A (all-todos) copia este fichero y todos los siguientes

Q (quit-abandonar) abandona sin copiar ni Éste ni ningún otro fichero

Si el fichero de destino existe ya, se requerirá al usuario para que pulse:

Y (yes-si) copia este fichero, sobregrabando el antiguo

N (no) no, no copies este fichero

A (all-todos) sobregraba el antiguo fichero, y cualquier otro fichero que vaya a ser copiado

Q (quit-abandonar) abandona sin copiar ni éste ni ningún otro fichero

Por ejemplo, si el directorio de omisión para datos es flp2_, y el de destino es flp1_:

WCOPY copia todos los ficheros de flp2_ a flp1_

WCOPY flp1_,flp2_ copia todos los de flp1_ a flp2_

WCOPY fred copia flp2_fred a flp1_fred,

flp2_freda_list a flp1_freda_list, etc.
WCOPY fred,mog copia
flp2_fred a flp2_mog,
flp2_freda_list a flp2_moga_list,etc.
WCOPY _fred,_mog copia
flp2_fred a flp2_mog,
flp2_frada_list a flp2_moga_list,
flp2_old_fred a flp2_old_mog,
flp2_old_freda_list a flp2_old_moga_list, etc
WCOPY _list,old_ copia
flp2_jo_list a flp2_old_jo_list,
flp2_freda_list a flp2_old_freda_list
WCOPY old__list,flp1_ copia
flp2_old_jo_list a flp1_jo_list
flp2_old_freda_list a flp1_freda_list

5.5.3 Copiando en Multitarea

El comando SPL copia ficheros de la misma manera que COPY_O, con la difererencia que lo que realmente hace es iniciar un programa en Multitarea
SuperTOOLKIT II CARE ELECTRONICS - QJUMP -20-
(llamado "SPooLer") que es el que hace el trabajo de copia, mientras que el intérprete de Basic queda libre para dedicarse a otras tareas. El comando SPL está pensado fundamentalmente para copiar ficheros a un impresora. Como una opción, se puede enviar a la impresora un salto de página (<FF>, caracter decimal 12, hexagesimal 0C) al final del fichero.

SPL nombre TO nombre vuelca un fichero
SPLF nombre TO nombre vuelca un fichero y añade un salto de página <FF> al final
El uso normal de este comando es con un solo nombre:
SPL_USE ser2 selecciona ser2 como valor de omisión para el "SPooLer"
SPLF fred vuelca fred a ser2, añadiendo
un salto de página al fichero

Cuando se usa este comando de esta manera, si el dispositivo de omisión está en uso, la tarea será suspendida hasta que el dispositivo se encuentre disponible. Esto significa que es posible volcar varios ficheros a la impresora a la vez.

Una variación en el uso de los comandos SPL y SPLF es utilizar canales del SuperBasic en lugar de nombres de ficheros. Estos canales deben abrirse antes de invocar al "SPooLer".

SPL #canal1 TO #canal2

donde canal1 y canal2 deben haber sido abiertos para entrada y salida respectivamente.

5.5.4 Renombrando Ficheros

Renombrar un fichero es un proceso similar a copiarlo, con la diferencia que no le se mueve ni se le duplica, sino que sólo se le cambia el nombre en el directorio. Los comandos son, sin embargo, exactamente iguales a sus equivalentes COPY.

RENAME nombre TO nombre ver COPY
WREN #canal,nombre TO nombre ver WCOPY

---------------------------------------------------

6 Programas en SuperBasic

En el Toolkit II se han redefinido todos los comandos para cargar, salvar y correr programas en SuperBasic. Las diferencias estriban en:

a) nombres de ficheros de omisión.
b) WHEN ERROR (sólo ROMs JS y MG).
c) utilización de la pila común.

6.1 DO

DO es un procedimiento adicional que permite la ejecución de comandos en SuperBasic desde un fichero.

DO nombre hace los comandos escritos en el fichero

Por ejemplo, el contenido de un fichero "seleccionar_impresora" podría ser:

OPEN #3,ser1: PRINT #3, CHR$(27);"C";"H":CLOSE #3

Este fichero fijará una impresora compatible EPSON a la longitud estándard europea de 72 líneas por página.

Si asumimos que el fichero "fijar_impresora" está almacenado en el directorio actual, en el dispositivo de omisión para datos, podrás fijar la longitud de página de la impresora con sólo decir:

DO elegir_impresora

Los comandos deben ser del tipo "directo": si hubiera alguna línea numerada, ésta sería añadida al programa SuperBasic actual. El fichero no debe contener ninguno de los comandos listados en esta sección (RUN. LOAD, etc), CONTINUE,

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -21-

RETRY o GOTO. Creemos que DO puede invocar procedimientos del SuperBasic sin ningún efecto perjudicial.

Un fichero DO puede contener claúsulas en línea:

FOR i=1 TO 20: PRINT "Esto es un fichero DO"

Si hay un comando RUN en un fichero DO, el fichero no será cerrado, una vez leído su contenido. Tampoco será cerrado, si es cargado con MERGE en lugar de con DO.

6.2. Directorios de Omisión

La mayoría de los comandos usan el directorio de omisión de datos. Además, el comando LOAD tratará de cargar el programa desde el directorio de omisión para programas en el caso en que no lo haya encontrado en el de datos.

6.3. Problemas con WHEN ERROR

El código para la manipulación de errores en las ROMs JS y MG tiene un problema ya que, una vez que se ha seleccionado WHEN ERROR, nunca es deseleccionado, ¡ aunque se elimine la cláusula WHEN ERROR mediante un NEW o un LOAD !. Todos los comandos de esta sección ponen a cero la señal del WHEN ERROR, y todos excepto STOP, ponen a cero el puntero a la cláusula WHEN ERROR actual.

6.4. Pila Común

El Toolkit II permite la reserva de espacio en la pila común. Este espacio es limpiado mediante los comandos que limpian las variables del SuperBasic: LOAD, LRUN, NEW y CLEAR.

6.5. Sumario de Comandos

DO nombre hace los comandos de un fichero

LOAD nombre carga un programa en SuperBasic

LRUN nombre carga y ejecuta un programa en SuperBasic

MERGE nombre une un programa en SuperBasic

MRUN nombre une y ejecuta un programa en SuperBasic

SAVE nombre,rangos salva un programa en SuperBasic

SAVE_O nombre,rangos como SAVE pero sobregraba el fichero, en el caso de que ya exista

RUN número de línea empieza un programa en SuperBasic

STOP para un programa en SuperBasic

NEW limpia el programa y las variables del SuperBaic

CLEAR limpia las variables del SuperBasic

--------------------------------------------------

7 Cargar y Salvar

El Toolkit II posee las mismas operaciones para cargar y salvar ficheros binarios que el QL estándard. Las diferencias son que las operaciones de salvado pedirán permiso para sobregrabar el fichero en el caso en el que exista, y que

todos los comandos usan los directorios de omisión.
Además hay dos variantes de las operaciones de salvado que permiten sobregrabar, y también el nuevo comando LRESPR.
LRESPR antes de cargar el fichero, lo abre, encuentra su longitud, y le reserva el espacio correspondiente en el área de procedimientos residentes. Una vez cargado, se hace un CALL a la dirección de comienzo del fichero.
El propio procedimiento CALL ha sido rectificado para evitar los problemas

que ocurren en las ROMs AH y JM cuando se hace un CALL desde programas de más de 32 Kbytes.

LRESPR nombre carga un fichero en el área de procedimientos residentes y hace un CALL
LBYTES nombre,dirección carga un fichero en la posición de memoria especificada
CALL dirección,parámetros llamada a código máquina
SBYTES nombre,dirección,tamaño salva un área de memoria
SBYTES_O nombre,dirección,tamaño como SBYTES pero sobregraba el fichero en el caso de que exista
SEXEC nombre,dirección,tamaño,datos salva un área de la memoria en forma de fichero ejecutable
SEXEC_O nombre,dirección,tamaño,datos como SEXEC pero sobregraba el fichero en el caso de que exista

Para SEXEC y SEXEC_O, el parámetro "datos" es el espacio de omisión para datos que requiere el programa.
Si hay alguna tarea en el QL (a parte del intérprete de SuperBasic o tarea 0), LRESPR dará el error "no finalizado". En ese caso, utiliza RJOB para eliminar todas las demás tareas.

--------------------------------------------------

## 8 Ejecución de Programas

El procedimiento que inicia la ejecución de programas ejecutables puede ser invocado mediante cinco comandos: EX y EXEC (que son sinónimos), EW y EXEC_W (que son sinónimos), y ET. Las diferencias son muy pequeñas: mientras que EX vuelve al SuperBasic en cuanto se inicia el programa, EW espera a que termine el programa antes de volver al SuperBasic. ET instala el programa pero vuelve el SuperBasic, de manera que se pueda trazar su ejecución mediante un programa monitor. Usaremos EX para describir todos los comandos.

### 8.1 Ejecución de Programas Individuales

En su forma más sencilla, EX puede ser utilizado para iniciar un único programa:
EX nombre
Se carga en el área de programas transitorios el programa contenido en el fichero "nombre" y se empieza su ejecución. Si el fichero no contiene un fichero ejecutable, aparecerá el error "parámetro incorrecto".
Es también posible pasar parámetros a un programa a través de una expresión de cadena:
EX nombre;parámetro de cadena
En este caso, se carga en el área de programas transitorios el fichero "nombre", se introduce la cadena en su pila y se comienza su ejecución.
Finalmente es posible para EX abrir ficheros de entrada y salida para un programa, además de (o en lugar de) pasarle parámetros. Si se prefiere, se puede pasar un número de canal del SuperBasic en lugar de un nombre de fichero. Un canal usado de esta forma debe de estar ya abierto.
EX nombre_del_programa,nombres de ficheros o #canal;parámetro de cadena
Tomando como ejemplo un programa UC que convierte un texto en mayúsculas, el comando
EX uc,fred,#1
cargará e inicializará el programa UC, que tiene a "fred" como fichero de entrada, siendo enviada la salida a la ventana #1.

8.2 Filtros

Se ha diseñado EX para seleccionar filtros para el proceso de un flujo de datos.

Es posible tener en el QL una cadena de tareas colaborando en el proceso de los mismos datos como si fuera una cadena de producción. Cuando se usa una producción en cadena, cada tarea realiza una parte bien definida del proceso total. La primera tarea tomará los datos originales y hará su parte del proceso; los datos parcialmente procesados serán pasados a la siguiente tarea la cual realizará su parte del trabajo; y así los datos pasarán gradualmente por todas las etapas del proceso. Los datos pasan de una tarea a la siguiente a través de un conducto llamado "pipe". Los datos reciben el nombre de "flujo" y las tareas que realizan el proceso de los datos se llaman "filtros".

Usando los símbolos [] para indicar un ítem opcional único y {} para representar una opción que puede ser repetida, la forma completa de EX será:

EX [#canal TO] espec_prog {TO espec_prog} [TO #canal]

donde espec_prog (especificación del programa) es

nombre del programa {,nombre de fichero o #canal} [;parámetro de cadena]

Cada separador TO crea un "pipe" entre tareas.

Todos los nombres y los parámetros de cadena pueden ser nombres, cadenas o expresiones de cadena. El significado de los nombres de fichero es, hasta cierto punto, dependiente del programa; hay, sin embargo, dos reglas que deben ser seguidas para todos los filtros:

1) la entrada principal de un filtro es el "pipe" que le une con la anterior tarea de la cadena (si existe), o, en caso contrario, el primer fichero de datos.

2) la salida principal de un filtro es el "pipe" a la siguiente tarea de la cadena (si existe) o el último fichero de datos.

Algunos filtros pueden tener solamente dos canales para entradas y salidas: la entrada principal y la salida principal.

Si el primer parámetro de EX es #canal TO, se cerrará el correspondiente canal del SuperBasic (si estaba abierto) y se reabrirá un nuevo canal como un conducto ("pipe") al primer programa.

Cualquier cadena enviada a este canal (por ejemplo, mediante PRINT) será procesada por la cadena de tareas. Cuando se cierre el canal, las tareas serán eliminadas del QL.

Si el último parámetro de EX es TO #canal, se cerrará el correspondiente canal de SuperBasic (en el caso de que estuviera abierto) y se abrirá un nuevo canal que servirá de conducto ("pipe") al último programa.

Cuaquier dato que pase a través de la cadena de tareas, llegará a este canal, de manera que podrá ser leído (por ejemplo, mediente INPUT). Cuando todos los datos hayan pasado, las tareas se autoeliminarán y cualquier intento de lectura desde el canal dará el error de "fin de fichero". Se puede usar la función EOF para averiguar si el fichero ha terminado.

8.3 Ejemplo de Proceso mediante Filtros

Como un ejemplo de proceso mediante filtros, encadenemos los siguientes programas para que procesen un único fichero: "uc" convierte un fichero a mayúsculas, "lno" numera las líneas de un fichero, y "page" lo divide en páginas con un encabezamiento opcional.

EX uc,fred TO lno TO page, ser;'Fichero fred, a '&DATE$

El filtro "uc" convierte el fichero "fred" a mayúsculas y lo pasa a través de un conducto a "lno". Este, a su vez, añade los números de línea y envia el resultado al conducto que lo une con "page". Finalmente "page" divide el fichero en páginas, añadiendo al principio de cada una el encabezamiento 'Fichero fred, a' seguido de la fecha, antes de enviar el fichero a la puerta "ser". Obsérvese que el fichero "fred" no es modificado; las versiones modificadas son simplemente transitorias.

---------------------------------------------------

9 Control de Tareas

Como el QDOS es un sistema operativo multitarea, es posible tener a la vez un número de tareas compitiendo o colaborando. Compitiendo por los recursos en



función de su prioridad, y colaborando comunicándose mediante el uso de conductos ("Pipes") o de memoria compartida. Los atributos básicos de una tarea son su prioridad y su posición en el árbol de tareas (su propietaria). Para identificar una tarea se utilizan dos números: uno es el número de tarea (que es

su índice en la tabla de tareas) y el otro un contador que sirve para diferenciar la tarea de otra eliminada que hubiera ocupado anteriormente la misma posición en la tabla de tareas. Los dos números combinados dan la identificación de la tarea (id) que es igual al número de tarea + contador*65536. Las siguientes rutinas de control de tareas utilizan dicho número de identificación, el cual puede ser dado como un sólo número (bien calculado de acuerdo a la fórmula anterior, bien tal como es devuelto por las funciones OJOB y NXJOB del Toolkit II) o como un par de números (número de Tarea, contador). Es decir, el parámetro único 65538 (2+1*65536) es equivalente a los dos parámetros 2,1.

9.1 Comandos para el Control de Tareas

JOBS es el comando que lista todas las tareas que se encuentran ejecutando en el QL en un momento determinado. Si el número de tareas que hay en la máquina es mayor que el que puede ser listado en la ventana, el prodecimiento congela la salida (CTRL F5) cuando la ventana se llena. El procedimiento puede fallar si se elimina una tarea del QL mientras está listándolas. Para cada tarea se da la siguiente información:

El número de tarea
El número de contador para la tarea
El número de la tarea propietaria de la tarea
La señal "s" si la tarea está suspendida
La prioridad de la tarea
El nombre de la tarea (o programa)

El comando es:

JOBS lista las tareas actuales en #1
JOBS #canal lista las tareas actuales en #canal
JOBS nombre lista las tareas en "nombre"

Tres son los procedimientos que permiten el control de tareas en el QL:

RJOB id o nombre,código de error elimina una tarea
SPJOB id o nombre,prioridad selecciona la prioridad de una tarea
AJOB id o nombre,prioridad activa una tarea

Si en lugar de una identificacióná, se da el nombre de la tarea, el procedimiento buscará la primera tarea que pueda encontrar con el nombre dado.

Si existe alguna tarea esperando que termine otra eliminada con RJOB, será liberada con un valor de D0 igual al código de error dado.

Por ejemplo:

RJOB 3,8,1 elimina la tarea 3,8 con error -1
SPJOB demon,1 selecciona a 1 la prioridad de la tarea llamada "demon"

9.2. Funciones sobre el Estado de las Tareas

Las funciones sobre el estado de las tareas se incluyen para posibilitar la exploración desde el SuperBasic del árbol de Tareas, de forma que se puedan realizar procedimietos complejos de control de tareas.

PJOB (id o nombre) devuelve la prioridad de la tarea
OJOB (id o nombre) devuelve la propietaria de tarea
JOB$ (id o nombre) devuelve el nombre de la tarea

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -25-

NXJOB (id o nombre,id de la tarea superior) devuelve la siguiente tarea del árbol

NXJOB es una función más bien compleja. El primer parámetro es la identificación de la tarea que está siendo examinada actualmente. El segundo parámetro es la identificación de la tarea superior del árbol. Si el primer parámetro pasado corresponde a la última tarea que pertenece, directa o indirectamente, a la tarea superior, NXJOB devolverá el valor 0; de lo contrario, devolverá la identificación de la siguiente tarea del árbol.

La tarea 0 siempre existe y es propietaria, directa o indirectamente, de todas las tareas del QL. Por ello, una exploración que empiece con los

parametros (0,0), explorará todas las tareas del QL.
Es posible que termine una tarea durante la exploración del árbol. Como precaución contra este hecho, las funciones de estado de las tareas devuelven los siguientes valores cuando son llamadas con valores inválidos de identificación de tareas:

PJOB = 0 OJOB = 0 JOB$ = "" NXJOB = -1

---------------------------------------------------

10 Abrir y Cerrar

Todos los comandos y funciones OPEN y CLOSE evitan el problema que ocurre usando las facilidades estándard del QL cuando se abren más de 32768 ficheros en una sesión de trabajo.

10.1 Comandos de Apertura
Los comandos OPEN del QL estándard han sido modificados para que usen los directorios de omisión para datos. Se han añadido dos comandos que permiten la apertura de ficheros sobregrabando el antiguo en el caso de que exista, y la apertura de un directorio.

OPEN #canal,nombre abre un fichero
OPEN_IN #canal,nombre abre un fichero sólo para lectura
OPEN_NEW #canal,nombre abre un nuevo fichero
OPEN_OVER #canal,nombre abre un nuevo fichero, y lo sobregraba si existe
OPEN_DIR #canal,nombre abre un directorio

10.2 Estado de los Ficheros
La función FTEST determina el estado de un fichero o dispositivo. Abre un fichero sólo para lectura y lo cierra inmediatamente. Si el fichero existe devolverá los valores 0 o -9 (código de error "en uso"); si no existe, devolverá -7 (código de error "no encontrado"). Otras posibles respuestas son -11 ("nombre incorrecto"), -15 ("parámetro incorrecto"), -3 ("sin memoria") o -6 ("canal no abierto" por no haber sitio en la tabla de canales).

FTEST (nombre) verifica el estado del fichero

La función puede usarse para comprobar que un fichero no existe:

IF FTEST (file$) = -7: Print "Fichero ";file$;" existe"

10.3 Funciones para la Apertura de Ficheros
Hay un grupo de funciones para la apertura de ficheros. Estas funciones difieren de los procedimientos OPEN en dos cosas. La primera es que si ocurre un error en el sistema de ficheros (por ejemplo, "no encontrado" o "ya existe"), estas funciones devolverán el número de error y el programa podrá continuar. La segunda diferencia es que estas funciones pueden ser utilizadas para encontrar un hueco en la tabla de canales; en este caso, si el proceso de apertura acaba positivamente, el número devuelto será el del canal.

FOPEN (#canal,nombre) abre un

fichero para lectura/escritura
FOP_IN (#canal,nombre) abre un fichero sólo para lectura
FOP_NEW (#canal,nombre) abre un fichero nuevo
FOP_OVER (#canal,nombre) abre un fichero nuevo, y lo sobregraba si ya existe
FOP_DIR (#canal,nombre) abre un directorio

Cuando se llama a estas funciones con dos parámetros, devuelven un valor de cero si son completadas correctamente, o el código de error negativo, en caso contrario.
Un fichero puede ser abierto sóla para lectura con una extensión opcional, usando el código siguiente:

ferr=FOP_IN(#3,nombre$&"_ASM") : REMark trata de abrir el fichero "_ASM"
IF ferr=-1:ferr=FOP_IN(#3,nombre$) : REMark en caso de error "no encontrado" trata de abrir fichero no "_ASM"

El parámetro #canal es opcional: si no se da, las funciones buscarán un canal libre en la tabla de canales, y si se llega a abrir el canal, su número será devuelto por la función. Observa que los números de error son siempre negativos y que los números de canal son siempre positivos.

En el ejemplo:

```
numcan = FOP_NEW(fred)                : REMark abre fred
IF numcan<0 : REPORT numcan : STOP : REMark fracasó open
PRINT #numcan, "Este es el fichero Fred"
CLOSE #numcan
```

no hay necesidad de conocer el número real del canal.

10.4 Cerrar

El comando CLOSE admite varios parámetros. Además, si se le llama sin ningún parámetro, cerrará los canales número #3 y superiores. No da ningún error si el canal no está abierto.

CLOSE #canales cierra canales

Por ejemplo

CLOSE #3,#4,#6 cierra #3,#4 y #6

---------------------------------------------------

11 Información de Ficheros

Seis funciones leen la información de la cabecera de un fichero.

Si un fichero está siendo ampliado, su longitud podrá ser determinada mediante la función FPOS que devuelve la posición actual del puntero del fichero (en caso de necesidad, el puntero del fichero puede ser llevado al final del fichero mediante el comando GET #n\999999).

FLEN (#canal) devuelve la longitud del fichero

FTYP (#canal) devuelve el tipo de fichero

FDAT (#canal) devuelve el espacio para datos del fichero

FXTRA (#canal) devuelve la información extra del fichero

FNAME$(#canal) devuelve el nombre del fichero

FUPDT (#canal) devuelve la fecha de actualización del fichero

El tipo de fichero es 0 para

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -27-

ficheros ordinarios
1 para programas ejecutables
2 para código máquina relocalizable

Estas funciones pueden ser también usadas con canales implícitos. Por ejemplo:

PRINT FLEN (#3) imprime la longitud del fichero abierto en el canal #3

PRINT FLEN (\fred) imprime la longitud de fichero "fred"

---------------------------------------------------

12 Acceso Directo a los Ficheros

En el QDOS, los ficheros apararecen como un flujo continuo de bytes. En dispositivos de directorio (Microdivres, discos duros, etc), el puntero del fichero puede ser colocado en cualquier posición del fichero. De esta forma, es posible el acceso directo a cualquier dato almacenado en el fichero. Acceso implica tanto acceso para lectura como, siempre que el fichero no haya sido abierto sólo para lectura, acceso para escritura. Se pueden escribir a y leer desde cualquier posición del fichero partes tan pequeñas como un sólo byte. El

QDOS no impone ninguna estructura fija para el registro en los ficheros: las aplicaciones pueden hacerlo, si lo desean.

Se dan procedimientos para el acceso a bytes únicos, a números enteros y en coma flotante, y a cadenas. Hay también una función para encontrar la posición actual en el fichero.

Con el fín de mantener en orden los ficheros, hay un comando que permite truncarlos (cuando ya no se requiera la información de la parte final del fichero), y otro para nivelar los amortiguadores ("buffers") del fichero.

En los comandos de acceso directo para entradas y salidas (E/S) se debe especificar el canal de entrada/salida, el puntero a la posición en el fichero en la cual comenzará la operación de entrada/salida, y la lista de ítems que entrarán o saldrán.

COMANDO #canal\posición,ítems

Normalmente se especificará el número de canal para los comandos de E/S directa, siendo, en caso contrario, #3 el valor de omisión. Si no se da la posición, las rutinas leerán o escribirán a partir de la posición actual; de lo contrario, se busca la posición dada antes de procesar la lista de ítems de E/S. Si el puntero es una variable en coma flotante en lugar de una expresión, una vez efectuada la lectura/escritura en el fichero, el valor de dicha variable será actualizado a la nueva posición actual del puntero en el fichero. Si no se da ningún ítem, no se escribirá/leerá nada en el fichero; en este caso, el comando puede ser usado para posicionar el puntero en un lugar determinado del fichero para el uso de otros comandos (por ejemplo, INPUT para entradas formateadas).

12.1 Entradas/Salidas (E/S) de Bytes

BGET #canal\posición,ítems lee bytes de un fichero

BPUT #canal\posición,ítems escribe bytes en un fichero

BGET lee 0 ó más bytes de un canal. BPUT escribe 0 ó más bytes en un canal. Para BGET, los ítems deben ser variables de coma flotante o enteras; para cada variable se coge un byte del canal. Para BPUT, los ítems deben ser variables que puedan ser evaluadas a un número entero entre 0 y 255; por cada ítem se envía un byte al canal de salida.

Por ejemplo, las sentencias

```
abcd=2.6
zz%=243
BPUT #3, abcd+1,"12",zz%
```

escribirán los bytes/caracteres correspondientes a los valores 4, 12 y 243 a continuación de la posición actual del puntero en el fichero abierto como canal

SuperTOOLKIT II CARE ELECTRONICS - QJUMP -28-

#3.

Siempre y cuando no se intente seleccionar la posición del puntero del fichero, las rutinas de E/S directa pueden usarse también para enviar de forma no formateada datos a dispositivos que no formen parte del sistema de ficheros. Así, si se abre el canal #3 a una impresora compatible EPSON, se podrá seleccionar la letra condensada-subrayada, bien con

```
BPUT #3,15,27,45,1
```

o con

```
PRINT #3,chr$(15);chr$(27);"-";chr$(1);
```

¿Cuál es más fácil?

12.2 Entradas/Salidas (E/S) No Formateadas

Es posible leer y escribir valores en su formato interno. Los comandos PRINT e INPUT del SuperBasic utilizan E/S formateadas, mientras que las rutinas GET y PUT de E/S directa utilizan E/S no formateadas. Por ejemplo,

```
PRINT 1.5
```

imprimirá los bytes 49 ("1"), 46 (".") y 53 ("5"). Sin embargo, dicho número es representado internamente mediante 6 bytes (igual que todos los números en coma flotante). Estos seis bytes tienen el valor 08 01 60 00 00 00 (en hexagesimal).

```
PUT 1.5
```

enviará los seis bytes anteriores al canal de salida.

El formato interno de un número entero es de dos bytes (siendo el primero el más significativo). El formato interno de un número en coma flotante es de 6 butes: los dos primeros representan el exponente (al que se le ha sumado el valor hexagesimal $81F) y los cuatro siguientes la mantisa (normalizada de forma que los dos bits más significativos -el 31 y el 30- sean diferentes) de la representación del número en base 2. Una cadena se representa internamente

mediante sus caracteres precedidos por dos bytes que indican la longitud de la cadena.

GET #canal\posición,items lee de un fichero datos en el formato interno

PUT #canal\posición,items escribe en un fichero datos en el formato interno

En GET, cada ítem debe ser una variable ya sea entera, en coma flotante o de cadena. Cada ítem debe corresponderse con el tipo de dato escrito en el fichero. En PUT, el tipo de dato escrito en el canal es el tipo del ítem en la lista de parámetros. Los comandos

```
puntofich=54
....
num%=42:salario=78000:nombre%="Smith"
PUT #3\puntofich,num%,salario,nombre$
```

posicionarán el puntero en el byte 54 del fichero abierto en el canal 3, y escribirán 2 bytes correspondientes al número entero 42, 6 bytes correspondientes al número en coma flotante 78000, y 7 bytes (dos que indicarán la longitud de la cadena y sus cinco caracteres) correspondientes a "Smith". La variable "puntofich" tomará el nuevo valor de 69 (=54+2+6+7).

Contrariamente al caso de variables o elementos de matrices, una expresión puede ser, a veces, escrita en diferentes tipos de formato. Se puede forzar un tipo concreto con uno de los siguientes trucos:

```
expresión +0     fuerza el tipo de coma flotante
expresión &""    fuerza el tipo de cadena
expresión ||0    fuerza el tipo entero
```

Así

```
xyz$="ab258.z"
....
PUT #3\37,xyz$(3 to 5)||0
```

posicionará el fichero abierto en el canal 3 en su byte 37 y escribirá el entero 258 en el fichero en la forma de dos bytes de valor 1 y 2 (ya que



258=1*256+2).

12.3 Truncado de un Fichero

TRUNCATE #canal\posición trunca un fichero

Si no se da una posición, el fichero es truncado en la posición actual.

TRUNCATE #dbcan

truncará el fichero abierto en el canal "dbcan".

12.4 Nivelado de Amortiguadores

FLUSH #canal nivela los amortiguadores del fichero

En el QDOS, los controladores de dispositivo de tipo directorio mantienen la mayor parte posible de un fichero en RAM. Por ello, un fichero puede quedar en una forma incompleta en caso de un fallo de la corriente o de cualquier otro accidente. El comado FLUSH ("nivelar") permite la actualización de un fichero sin necesidad de cerrarlo. El cierre de un fichero causa siempre su nivelado. El Toolkit II actualiza las rutinas de los microdrives de manera que puedan efectuar un nivelado completo. Sin embargo, FLUSH no trabajará con los sistemas de disco MicroPeripherals a no ser que hayan sido actualizados a la versión QFLP.

12.5 Posición de un Fichero

La función FPOS ayuda al trabajo con ficheros en acceso directo ya que devuelve la posición actual del puntero del fichero. Su sintaxis es:

FPOS(#canal) devuelve la posición del fichero

Por ejemplo:

```
PUT #4,102,valor1,valor2
ptr=FPOS(#4)
```

asignará a "ptr" el valor 114 (=102+6+6).

La posición del puntero del fichero puede ser seleccionada mediante los comados BGET, BPUT, GET o PUT sin ningún item para leer o escribir. Si se intenta poner el puntero más allá del final del fichero, éste será colocado al final, no devolviéndose ningún mensaje de error. Obsérvese que la selección de

la posición del puntero no significa que dicha parte del fichero esté realmente en un buffer, sino que dicha parte está siendo buscada. De esta forma es posible que una aplicación controle la prebúsqueda de partes de un fichero en las que el controlador de dispositivo es capaz de prebúsqueda.

---

---

OTROS TEMAS

---

LA ERA DE SIR CLIVE SINCLAIR

(Este pequeño artículo lo escribí hace un año. El microdrive en que estaba grabado hizo la gracia de mandar a freir espárragos todo lo que tenía dentro, con ese encanto especial que tienen los microdrives para mandar sus ficheros a freir espárragos... afortunadamente, había sacado una copia impresa, copia que ¡por fin! me he decidido a teclear de nuevo:)

Cuando releo publicaciones de informática de hace unos años, me llaman la atención los anuncios comerciales en los que se anuncian ordenadores o interfaces a precios muy superiores al que hoy tienen equipos equivalentes o más potentes. También es interesante fijarse en el tipo de ofertas de los anuncios de ocasión. Hace cinco años los anuncios más abundantes de ese tipo eran para buscar comprador a un "ZX-81 en perfecto estado"; ahora, sin embargo, ocupan su lugar las ofertas de ZX Spectrums o QLs. Bien, inmersos en este "paraiso" consumista, es comprensible que la mayoría de los usuarios consideren a su máquina como de "usar y tirar" y añadan pretenciosamente al texto del anuncio la muletilla "por cambio de equipo". ¿Pero qué es lo que ocurre exactamente? Entiendo que se pueda desear el adquirir un ordenador más potente, y que un usuario de ZX-81 quisiera en el año 1983 deshacerse de él cuanto antes, pero es triste el desprecio tan silencioso que se hace los ordenadores cuando pasa un tiempo, la utilización comercial que se hace de ellos alrededor de la fecha de su lanzamiento y la muerte que se les provoca cuando ya no pueden producir más dinero. Como a las lavadoras.

Imagino que, entre los usuarios del QL, una buena parte hemos compartido con anterioridad mucho tiempo junto a un Spectrum, del mismo modo que buena parte de los antiguos amantes del Spectrum lo fueron en su día del ZX-81 o incluso del ZX-80. No sólo eso, sino que me confieso mentalmente incapacitado para desprenderme de mis (dos) Spectrum, e incluso me atrevería a afirmar que en algún lugar debe existir alguien más capaz de entender el apego que se puede llegar a sentir por una pizca de silicio calenturiento.

Estamos en el final de una época para los microordenadores. Una época en cuyo inicio tuvo mucho que ver Sinclair. Ahora se apaga moribunda la explosión de comienzos de los años 1980 y penetramos en una nueva explosión de diferentes características, pero ya de mano de la experiencia; conservando la capacidad de dejarnos maravillar que aprendimos entonces, pero sin la inocencia de aquellos años.

El cambio es evidente casi con sólo hojear cualquier publicación de informática de hace cinco, cuatro, e incluso tres o dos años. Empezando por la oferta, los precios, y concluyendo en las consultas de los lectores. En aquellas fechas los micros de 8 bits se apoderaban por asalto de los hogares y comenzaba la era Sinclair. En realidad podemos considerar el año 1980 como el inicio de la nueva era, ¡el año cero de nuestra particular cronología! Los primeros frutos del genio británico no tuvieron mucha repercusión en España, eran otros tiempos... fue hace mucho, mucho tiempo: hacia el año 1 d.S. (después de Sinclair). Pero en el año 2 d.S. nació el ZX Spectrum y el mundo jamás volvió a ser el mismo.

Por aquel entonces un micro de 16 Kbytes de RAM era algo estupendo, uno de 48 Kbytes era extraordinario, y uno de 64 Kbytes era increible. Proliferaban las publicaciones con listados en BASIC gracias a los cuales dábamos nuestros primeros pasos, era cosa común el pirateo de programas en cinta: primero copiándolos analógicamente de "cinta a cinta", más tarde con los novedosos programas copiadores...

No sería inconveniente comentar aquí, por ampliar algo el tema, que el

mismísimo año 1 d.S. IBM acababa de presentar su PC: unos "increibles" 64 Kbytes de RAM alrededor de un ya por entonces vetustillo microprocesador 8088... con esto quiero hacer notar que la distinción entre tipos de ordenadores es, en la mayoría de las ocasiones, meramente debida a intereses comerciales, y que cada uno tiene un valor propio "en sí" por encima de la categoría en que es clasificado. El PC se convirtió en estándar por motivos comerciales y

página 1

económicos, por la solidez de la empresa que lo sustentaba, pero no por las excelencias de la máquina como tal. Aunque en aquella época sí había diferencia evidente entre un PC y un micro de 8 bits como el Spectrum o el Commodore 64, la diferencia radicaba principalmente (aparte de las diferentes potencias de los respectivos microprocesadores, que en la práctica no es tal, pues han existido ordenadores muy potentes construidos sobre un Z-80, por ejemplo) en las interfaces de que carecían ( disco, RS232, Centronics...), carencia inevitable si se quieren mantener bajos los costes y entrar en el mercado doméstico. Ahora, por el contrario, los "super-micros" que se vislumbran dominantes en la nueva era, como el Commodore Amiga, el Atari ST, (así como uno que empieza por "Q" y termina por "L", con todos sus compatibles: Thor, Thor 20, Thor XVI, Atari ST QL-Emulator, ¡y los que quedan!...) le dan, hablando claro, mil y una vueltas a cualquier PC que se les ponga a tiro, en todos los campos... Y es que tanto el Amiga como el ST como el QL pueden emular mediante software a un PC, pero ¡lo contrario es imposible!

Alrededor de los años 3 y 4 d.S. comienza una feroz batalla por el dominio del mercado doméstico, que había empezado a saturarse. Primero, Sinclair "lava la cara" al Spectrum: le pone un teclado nuevo, muy similar al del ya nacido QL, y llama al vástago "Spectrum+". Poco después la batalla se dirige hacia el dominio de los 128 Kbytes: Commodore 128, Spectrum 128... Este último, apodado "Derby" (es fácil imaginar por qué) llega a tiempo de no perder el tren, pero no obtiene el respaldo tan mayoritario logrado por su predecesor de 48 Kbytes. En realidad, estas máquinas siguen siendo las mismas que sus antecesores, pero gracias a la paginación de bancos pueden acceder a más memoria, tanto ROM como RAM, e incorporan algunas otras nuevas facilidades, como interface RS232, MIDI, de joystick, y mejores capacidades sonoras. Concretamente el Spectrum 128, a pesar de que el precio no supera en mucho al del modelo anterior de 48 Kbytes, no tiene gran aceptación, y las empresas de software apenas se atreven a hacer programas especiales para él: sólo adaptan algunos de los ya existentes, añadiéndoles mejores efectos sonoros y poca cosa más.

¿Qué es lo que fallaba? Antes lo hemos mencionado de pasada: se trata del mismo ordenador. Es más, se trata de un desesperado intento de llevar al límite la tecnología de los 8 bits, de "estirar" demasiado el invento...

El aficionado medianamente inexperto lee en la propaganda "¡128 Kbytes!" y se imagina 128 Kbytes "uno detrás de otro". Luego compra el ordenador y se encuentra con la sorpresa de que no puede escribir programas en BASIC más extensos que en un modelo de 48 Kbytes... ¿Publicidad engañosa? Algo de eso sí hay, pero para que la publicidad pudiera hablar con mayor claridad, o para que, al menos, el comprador supiera lo que compra, sería imprescindible que todo potencial comprador de un microordenador supiese por qué un microprocesador Z-80 no puede manejar más de 64 Kbytes de una vez... ¡Pero, si todos los potenciales compradores de micros supiesen eso, sabrían además otras cosas igual de interesantes, y no comprarían un PC basado en un Intel 8088 u 8086, sino una máquina basada en un miembro de la familia Motorola 680XX!

Pues bien, hacia el año 4 d.S. vió la luz un nuevo miembro de la saga Sinclair, el cual, para no romper la tradición, se adelantó también a su tiempo... sólo que este tal vez se pasó un pelín: se trata del QL.

No es necesario comentar aquí las características que hacen del QL una máquina excepcional, puesto que todos las conocemos y por ello estamos en esto. Baste decir que hacia la fecha de su lanzamiento no existía otro ordenador, en su rango de precio, que pudiese hacerle la más mínima sombra.

Pero han pasado cinco años, y han aparecido otras máquinas que, si no más poderosas en términos globales, sí le alcanzan e incluso aventajan en algunos aspectos concretos, como son principalmente el sonido y los gráficos: tal es el caso del Commodore Amiga y el Atari ST. El QL se ha mantenido gracias al apoyo de un software de calidad, por parte de empresas muy concretas, y a la dedicación de sus seguidores, completamente abandonados a su suerte tras la compra de Sinclair Research por Amstrad.

Sería interesante que, quienes mantenemos vivo al QL con nuestra fidelidad, reflexionásemos detenidamente sobre las causas que nos han conducido a la tan particular situación en que nos encontramos. Es cierto que el QL tiene un montón de pequeñas pegas, defectos, errores... que -fue una metedura de pata- Sinclair nunca quiso reconocer, y es más cierto que la monumental canallada del señor

página 2

Sugar -alias "azucarillo" o "terroncete"- ha tenido mucho que ver. Pero es aún más evidente que el origen del problema está en que el QL se adelantó demasiado a su tiempo: podía haber aparecido perfectamente el año pasado, pero apareció hace cinco años... cuando cualquier cosa que no fuese un Z-80 o un 6502 sonaba a ciencia-ficción a los oidos del aficionado. Y cuando algo se adelanta demasiado a su tiempo, no es comprendido; y de la incompresión al desprecio dista un solo paso muy pequeño.

Habría que decir, no obstante, que la decisión de Amstrad de no continuar la producción y comercialización del QL, casi es de agradecer, desde cierto punto de vista: el señor "terroncete" se dió cuenta a todas luces de que el QL era muy superior a todas sus baratijas juntas (mis respetos a los usuarios de CPC y PCW), y decidió quitárselo de encima. Si no hubiese obrado así, es presumible que ahora el QL fuese otra de sus "ofertas increibles", otro numerito en sus previsiones de ventas anuales, que es en lo que ha convertido al Spectrum... Y es preferible estar desterrado que ser una pieza más de sus peseteros estudios de marketing.

Bien, parecía entonces que la era Sinclair tocaba a su fin, pero en el año 7 d.S., tras un periodo de inquietante silencio, el genial inventor nos presenta su última creación: el Z88, un portátil con pantalla LCD, basado en un Z-80 y con un sistema de almacenamiento externo mediante cartuchos EPROM y otras ROMs alterables y RAMs alimentadas con baterías. Sinclair nunca fue partidario de los discos, y la opinión general le recriminó su actitud cuando no se contentó con dotar de microdrives al Spectrum, sino que los instaló como estándar en el QL. Ahora no es necesario ser adivino para darse cuenta de que, en unos años, el uso de discos magnéticos irá decreciendo en favor de los discos ópticos, y que éstos después cederán paso a los cartuchos de memoria ROM alterable, porque al fin y al cabo todos los discos se basan en un dispositivo mecánico para su funcionamiento... y todo lo que se mueve puede dejar de hacerlo en el momento más inoportuno.

Pero mientras nosotros, pobres mortales, nos dedicamos a especular sobre si tal cambio se producirá dentro de cinco o de veinte años, Sir Clive Sinclair ya lo ha comenzado por su cuenta.

Marcos Cruz, Madrid, 27/3/1989

LEYES DE MURPHY AGRUPADAS

LEYES SELECTIVAS DE LA GRAVEDAD:

1.-Todo objeto caerá siempre del modo que produzca el mayor daño posible.

2.-De todos los objetos que pueden caerse, siempre caerá el más frágil, el más valioso, o el más delicado.

3.-Cuando un objeto se cae, siempre va a parar al lugar más inaccesible, después de golpear el punto más delicado o más doloroso.

4.-Las posibilidades de que la tostada con mantequilla caiga sobre la alfombra con el lado de la mantequilla hacia abajo, están en relación directa con el precio de la alfombra.

CONTRALEYES UNIVERSALES:

1.-El dolor de muelas siempre empieza el viernes por la tarde.

2.-En autopista, el otro carril siempre circula más rápido.

PRINCIPIO DE PETER:

*.-En una organización jerárquica, cada empleado alcanza un nivel que depende sólo de su grado de incompetencia.

LEY FUNDAMENTAL DE MURPHY:

*.-Si algo puede funcionar mal, funcionará mal.

PRINCIPALES COROLARIOS A LA LEY DE MURPHY:

1.-Si hay alguna posibilidad de que varias cosas no funcionen, la que deja de hacerlo es siempre la que causa un destrozo mayor.

2.-Si dejamos que las cosas vayan a su aire, tenderán por sí mismas a ir de mal en peor.

3.-Cada solución origina nuevos problemas, la madre Naturaleza se pone siempre del lado de los problemas.

4.-Sonría, mañana será peor.

5.-La materia se daña en razón directa a su precio.

6.-Todo deja de funcionar al mismo tiempo.

LEY DE CHISLOM:

página 3

*.-Cuando las cosas van bien, eso quiere decir que algo va mal. Si las cosas ya no pueden ir peor, irán peor.

LEY DE FINAGLE:

*.-Si un experimento sale bien, es porque algo ha funcionado mal. En cualquier operación, el dato que parece absolutamente correcto es el que está

absolutamente equivocado.

LEY DE SOOD:

*.-Antes o después, las circunstancias más adversas que podamos imaginar se producirán.

COMENTARIO DE HERMAN:

*.-¿Quién dijo que las cosas iban a ir mejor?

TEOREMA DE STOCKMAYER:

*.-Si algo parece fácil, es difícil; si parece difícil es imposible.

PRIMERA LEY DE JOHNSON:

*.-Cuando algo falla, lo hace siempre en el momento más inoportuno.

LEY DE YOUNG:

*.-Los grandes descubrimientos se hacen siempre por error.

LEY DE LOS PENSAMIENTOS NEGATIVOS:

*.-Si usted sigue una línea de pensamiento negativo, obtendrá resultados negativos; si usted sigue una línea de pensamiento positivo, obtendrá resultados negativos.

LEY DE CORNUELLES:

*.-La salud es un estado precario que no presagia nada bueno.

POSTULADO DE BOLING:

*.-Todo lo que es bueno, es ilegal, es inmoral, o engorda.

LEY DEL INNOMBRABLE:

*.-En cuanto mencionas algo, si es bueno, despídete de ello; si es malo, ocurrirá al instante.

LEY DE ISAWY:

*.-Un atajo es siempre el camino más largo entre dos puntos.

LEYES DE SEGAL:

1.-Un hombre con un reloj siempre sabe la hora que es, un hombre con dos relojes no está seguro jamás.

2.-Bien está lo que acaba sólo mal.

3.-Murphy era un optimista.

4.-Bienaventurados los que no esperan nada, ya que nunca resultarán decepcionados.

5.-La paloma siempre se caga encima de uno.

6.-No va a funcionar.

(Texto reproducido para CUQ por:)

Marcos Cruz, Madrid, 31/3/1989

----------------------------------------------------------------------

## LENGUAJE FORTH

----------------------------------------------------------------------

TRANS-BASE, UNA PEQUEÑA UTILIDAD EN FORTH.

En gran cantidad de ocasiones, cuando me siento ante el ordenador para intentar aprender algo sobre sus interioridades, me encuentro con la necesidad frecuente de realizar cambios de base numérica, fundamentalmente entre decimal, binario y hexadecimal, aunque en alguna ocasión, también he necesitado cambios a octal.

Cualquiera que haya manejado el Forth sabe lo sencillo que resulta realizar este tipo de cambios sobre la marcha cuando se está trabajando dentro del intérprete. Esta comodidad es la que solía hechar de menos cuando esta trabajando con Basic o con un monitor o simplemente echando un vistazo a algún listado en ensambladoro al libro de Adrian Dickens.

Dado que ninguna de las calculadoras al uso en el QL contienen esta posibilidad del cambio de base, me decidí a hacerme yo mismo una pequeña utilidad que me evitara tener que hacer los cambios con papel y lápiz o andar rebuscando entre tablas. Lógicamente elegí el Forth para ello y en esta ocasión me decidí por el Super-Forth de D.P. por variar.

El listado del programa es el siguiente:

```
( ****** Trans-Base  v 1.0    Abril 1989   José Carlos de Prada ******* )

          DECIMAL
          2VARIABLE #BASE10    ( variables para los id de los canales )
```

```
          2VARIABLE #BASE2
          2VARIABLE #BASE4
          2VARIABLE #BASE8
          2VARIABLE #BASE16

          31 STRING NUMERO          ( variable de cadena para entrada )
          2VARIABLE NUM            ( variable para entrada convertida )
          2.147483647  2CONSTANT  MAXIMO                ( valor máximo )

          0 OPEN CON_192X12A300X22 #BASE10 2!       ( abrir canales y  )
          0 OPEN CON_192X12A300X36 #BASE2  2!       ( guardar su id    )
          0 OPEN CON_192X12A300X50 #BASE4  2!       ( en las variables )
          0 OPEN CON_192X12A300X64 #BASE8  2!
          0 OPEN CON_192X12A300X78 #BASE16 2!

: #IO     ( d -- ) ( pone la entrada/salida en el canal guardado en d )
          2@ 2DUP  #IN 2!  #OUT 2! ;
: LETREROS ( -- )                               ( rotula los letreros )
          #DEFAULT 2DUP #IN 2! #OUT 2!
          CLS
          4  2 CURSOR ."  DEC"
          4 INK
          4 16 CURSOR ."  BIN"
          4 30 CURSOR ."  B-4"
          4 44 CURSOR ."  OCT"
          4 58 CURSOR ."  HEX" ;
: VENTANA  ( fondo papel col-borde ancho-borde tinta -- )
          INK BORDER PAPER STRIP CLS ;    ( parámetros de una ventana )
: VENTANAS ( -- )                   ( pone parámetros de las ventanas )
          LETREROS
          #BASE10    #IO 2 2   4 1 7 VENTANA
          #BASE2     #IO 4 4   2 1 0 VENTANA
          #BASE4     #IO 4 4   2 1 0 VENTANA
          #BASE8     #IO 4 4   2 1 0 VENTANA
          #BASE16    #IO 4 4   2 1 0 VENTANA  ;
: SALIDA  ( n ad -- )    ( escribe la salida en el canal ad en base n )
          #IO CLS NUM 2@ ROT BASE ! 31 D.R ;
: BN      ( n -- )          ( convierte una entrada al resto de bases)
          DECIMAL  BASE !
          CLS
          NUMERO INPUT                                ( lee la entrada )
          0. NUMERO CONVERT NUMERO COUNT + >=   ( conversión a número )
          IF
               2DUP MAXIMO 2SWAP D<           ( error de rebosamiento )
               IF
                    2DROP CLS ."           REBOSAMIENTO"
               ELSE
                    NUM 2!                            ( salida correcta )
                    10 #BASE10 SALIDA
                    16 #BASE16 SALIDA
                    8  #BASE8  SALIDA
                    4  #BASE4  SALIDA
                    2  #BASE2  SALIDA
                THEN
          ELSE                                 ( error de tipo de dato )
                CLS ."         ERROR DE ENTRADA"
          THEN
          DECIMAL ;
: B10     ( -- )                                ( entrada en base 10 )
          #BASE10 #IO 10 BN ;
: B2      ( -- )                                ( entrada en base  2 )
          #BASE2  #IO  2 BN ;
: B4      ( -- )                                ( entrada en base  4 )
          #BASE4  #IO  4 BN ;
: B8      ( -- )                                ( entrada en base  8 )
          #BASE8  #IO  8 BN ;
: B16     ( -- )                                ( entrada en base 16 )
          #BASE16 #IO 16 BN ;
: ELIGE                              ( selección de la base de entrada )
          KEY CASE
```

```
                     68 OF B10 0 ENDOF
                    100 OF B10 0 ENDOF
                     66 OF B2  0 ENDOF
                     98 OF B2  0 ENDOF
                     67 OF B4  0 ENDOF
                     99 OF B4  0 ENDOF
                     79 OF B8  0 ENDOF
                    111 OF B8  0 ENDOF
                     72 OF B16 0 ENDOF
                    104 OF B16 0 ENDOF
                     27 OF     1 ENDOF
                     DEFAULT 100 500 BEEP 0
          ENDCASE ;
: TRANSBASE   ( -- )                              ( programa principal )
          VENTANAS
          BEGIN
                     ELIGE
          UNTIL
          BYE ;
END_FILE TRANSBASE
```

El código fuente contiene bastantes comentarios entre paréntesis como para que resulte sencillo de comprender. La mayor parte del programa se dedica al manejo de canales y ventanas, siendo la palabra BN el núcleo del programa que se encarga de leer del teclado el número en la base de entrada y transformarlo al resto de bases (para que quede completo he elegido como bases la decimal y las potencias de 2 de 1 a 4).

Para una mejor presentación del programa y para su comodidad de manejo en multitarea se puede cargar automáticamente con EXEC FLP1_FORTH83_JOB, habiendo sustituido el bloque 1 (BLK1) por el que acompaña en el disco al fichero del código fuente y en el que se han modificado los parámetros de la ventana:

```
( Login block, called by COLD, modify as required )
0 OPEN CON_240x72a258x20   ( the main console window )
2DUP #IN 2! 2DUP #OUT 2!
#DEFAULT CLOSE ' #DEFAULT >BODY 2!
4 MODE 7 PAPER 7 STRIP 2 INK 0 0 CSIZE 255 1 BORDER
FLP1_
CLS       LOAD_FILE FLP1_TRANSBASE_FTH
```

La utilización del programa es bastante simple. Para seleccionar la base de entrada se pulsa la inicial del nombre de la base:

D --> Decimal
B --> Binario
C --> base Cuatro
O --> Octal
H --> Hexadecimal

cualquier otra pulsación dará como resultado un corto pitido.

Una vez seleccionada la base de entrada aparecerá el cursor en la ventana correspondiente. No se admiten números negativos, ni con coma o punto flotante. Cualquier valor caracter incorrecto producirá un "ERROR DE ENTRADA"; cualquier valor por encima de 2147483647 (Dec) dará un error de "REBOSAMIENTO". Para terminar el programa basta con pulsar ESC.

Espero que os resulte de utilidad.

José Carlos de Prada, Madrid, 23/4/89

```
*************************************************************

       /////////      ////   ////       ///////////
      IIIIIIIII/     IIII/  IIII/      IIIIIIIIIII/
     IIIII/          IIII/  IIII/             III/
     IIIII/          IIII/  IIII/           III/
```

```
IIIII/          IIII/  IIII/          III/
IIIII//////  // IIII///IIII/  //   III////////  //
 IIIIIIIII/ II/  IIIIIIIII/  II/  IIIIIIIIIII/ II/

*************************************************************
        CIRCULO   DE    USUARIOS    DE      Z88
*************************************************************
```

Complemento para Usuarios de Cambridge Z88

----------------------------------------------------------------------

JUNIO 1989

----------------------------------------------------------------------

... En el año 9 después de Sinclair, toda Hispania está siendo ocupada por las nuevas legiones de PCs portatiles ¿TODA? ¡NO! Unos misteriosos Z88 creados por Tio Clive Sinclair tienen la osadia de ponerlos en ridiculo cada vez que se ponen a tiro ....

----------------------------------------------------------------------

EDITORIAL

----------------------------------------------------------------------

CUZ como circulo de usuarios no existe actualmente como tal ya que solamente existe un usuario (yo mismo). Aún asi, se intentara informar sobre este sorpredente ordenador personal, que creo que podria ser de extrema utilidad a todo aquel que necesita usar un ordenador en situaciones en las cuales ni siquiera los portatiles o portables PCs pueden hacerlo.

S. MERINO

NOTICIAS Z88

Subscribirse a Z88 USER CLUB cuesta 12 meses : 6 libras UK, 9 libras Europa y 12 libras el resto del mundo. Contactar con TK COMPUTERWARE.

Magnetic Media, la firma que comercializa el Z88 Español, ofrece entre muchas cosillas el Z-TAPE, que es un sistema de almacenamiento en cintas de Cassette (13.495 ptas sin IVA).

TK COMPUTERWARE ofrece en su variado catalogo un Disk Drive 3.5" con interface y alimentado a pilas por 275 libras.

El Z88 solamente puede ampliarse y comunicarse con el exterior via RS-232, pero aunque el RS-232 fue criticado en el QL como un defecto (solamente por no haber elegido Centronics), gracias a ese conector se le puede conectar cualquier cosa (¡Hasta TRANSPUTERS!).

El Z88 ha bajado de precio en UK. Ahora cuesta 236 libras (¡Ha bajado un 20%!).

El Z88 desde su lanzamiento oficial en Europa ha visto su colección de software comercial aumentada a un minimo de 50 titulos conocidos en 6 meses (solamente en las listas de precios que yo poseo, deben haber muchisimos más).

También han aumentado considerablemente el número de libros dedicados al Z88 :

- Pipedream (Dabs)
- Z88 Computing (Ian Sinclair)
- Z88 Real Power Computing (P.R. Flaig)
- Z88 Reference Manual (M-Tec)
- Z88 Software Developers Guide (Cambridge Computers)
- Using your Z88
- Z88 Manual tecnico de Hard

El único que voy a comprar este verano es Z88 Software Developers Guide (cuesta 35 libras + gastos de envio).

Las diferencias entre una máquina basada en el Z80 y otra en el 8088-8086 (e 80286) son minimas. Los dos no pueden direccionar más de 64K y tienen que recurrir a la paginación o segmentación. Los dos poseen versiones que corren a velocidades similares. Y hay una gran cantidad de Hackers enamorados del Z80, y según ellos posee un buen juego de instrucciones.

Creo que después de todo lo que he visto, a las dos familias le quedan cuerda para rato (solamente hay que echar un vistazo al parque de máquinas instaladas). Aunque soy partidario de las ventajas que ofrece la familia 68000, me parece que solamente vamos a poder continuar disfrutandolas unos pocos elegidos (¡Qué le vamos a hacer!).

S. Merino

¿ POR QUE UN Z88 ?

Hace mucho tiempo, en 1979, mi ilusión siempre era comprar el más grande y potente que se podia permitir mi bolsillo. Pero la mayor disolusión siempre fue que el cacharro se quedaba pequeño o anticuado, y me costaba ampliarlo más que uno de los nuevos modelos. Otros inconvenientes : el software viejo no corria en la nueva máquina, el hardware (monitor, impresora, disco, cassette,...) no servia,....

Cuando me compré el QL estaba muy contento de disponer por primera vez de un ordenador de 32 bits con sistema operativo multitarea (aunque en aquellos dias no sabia usarlo), especialmente porque no habia ningún equipo de oficina que le hiciese sombra en aquella época. La otra cara del asunto era la falta de libros, software y hardware para el QL (una pequeña pega debida a su lanzamiento precoz).

Todo lo anterior sobre el QL, no tiene ni pizca de importancia si hablamos del jarro de agua fria que vino 6 meses después en Mayo de 1986.

Me habia gastado un cuarto de millón de ptas en un equipo condenado a morir antes de nacer, el CHOCK fue horroroso.

¿ Qué hacer ? Comprar un Atari ST (si lo hubiese hecho y me entero que Inves lo deja abandonado igual que el QL, no sé qué habria pasado), comprar un Amiga (precio prohibitivo, escasez de software y libros. Una locura comprarlo), comprar un Apple Macinstosh ( dificil de comprar uno en Málaga. Si lo hubiese hecho, seria incluso peor que continuar con el QL), comprar un PC (ni borracho),... Mi decisión fue continuar con el QL y esperar tiempos mejores. Por desgracia, los tiempos mejores ni se ven en el horizonte, y los PCs compatibles cada vez son más y más (se cuentan por millones, pero pocos programas originales se venden en ese parque de millones).

Estamos en 1989, y me estoy preguntando ¿Qué difrencia hay entre un ordenador vivo y uno muerto? Porque no conozco la respuesta. Pero creo que la decisión debe tomarla el usuario, pues es él quién debe decidir si esa máquina le gusta por tal o por cual (Hay gente que le gusta programar en FORTH y gente que no le gusta porque nunca le han enseñado o no lo han probado. También hay gente que le gusta programar bajo QDOS y gente que no han oido ni el nombre). En mi caso, durante todos estos últimos años he aprendido muchas cosas con mi QL (y más que supongo que me quedan por aprender). Creo que es uno de los mejores ordenadores que han sido fabricados y cada dia que pasa me gusta más (me va a ser casi imposible desprenderme de mis dos QLs).

Pero ¿Por qué comprar un Z88?

La idea de comprar un portátil siempre ha sido desde hace mucho tiempo una de mis mayores ilusiones.

Portatiles, que yo sepa, siempre han existido, pero eran auténticos monstruos o pequeños de muy baja potencia, capacidad de memoria y pantalla.

Hoy en dia cuando se habla de portátil, la gente piensa en uno de esos PC. El único problema que yo veo es que el MS-DOS no es el sistema operativo ideal para un portátil. La unidades de disco flexible y disco duro por su peso y consumo no son portatiles. El Tamaño de pantalla es quizás el ideal, pero el tamaño del aparato es otro cantar. Y lo peor de todo es la autonomia de unas 2 horas de media (sin contar el tiempo que hace falta para recargar las baterias recargables de algunos equipos). También hay que agregar el peso y volumen de los discos con programas y datos (los programas MS-DOS por alguna extraña razón son bastante más voluminosos en memoria RAM que en otros sistemas).

Vamos que podemos decir que un PC portátil, lo único que tiene de portátil es que es muy fácil de mover de un lado a otro en una oficina si hay toma de corriente cerca (indispensable si desea trabajar al menos 4 horas seguidas), o para llevarselo de viaje en el maletero e instalarlo en la habitación de un hotel. Eso si, hay que descubrirse, son más bonitos y elegantes que sus hermanos no portatiles.

Pero ¡TACHAN! llegó el Z88. Un ordenador portátil de 850 gramos, teclado QWERTY, pantalla LCD supertwist de 8 lineas X 106 caracteres, memoria RAM ampliable a 3 MB,.....

El Z88 utiliza 4 pilas normales o alcalinas de larga duración. Su autonomia sin ampliar es de 20 horas o un año en estado de reposo, pero será menor según la cantidad de cartuchos RAM y capacidad de las pilas.

Los cartuchos de RAM estatica consumen todos (32,128,512) casi la misma cantidad de energia, según manual Inglés.

Aunque no llevo el control del consumo de energia, hace unos 10 dias que lo estoy usando con un cartucho 128K RAM y todavía no he tenido que cambiarlas.

S. Merino, Fuengirola, 1/5/89

EL Z88 UNA MAQUINA MUY ESPECIAL

Es una máquina totalmente recomendable para aquellos que necesiten llevar su ordenador a cualquier sitio (es todo terreno). Quitando el Organiser que no es adecuado para procesar texto, es el único ordenador personal verdaderamente portátil (En UK ocupa el 20% del mercado y el segundo lugar en ventas; quedando el resto repartido entre los portables PC compatibles).

Sus posibilidades de comunicación con otros ordenadores y cualquier impresora son asombrosas (modem, telex, radio transfer,...). E incluso existe la posibilidad de acoplarle un monitor cuando exista el interface adecuado.

Funciona con 4 pilas de 1.5 voltios faciles de comprar en cualquier Supermercado e, incluso Kiosko callejero, que proporcionan 20 horas de actividad del ordenador o un año si está en reposo. El apagado coloca al Z88 en modo de reposo; todos los datos de RAM permanecen intactos, consumiendose un minimo de energia de las pilas del Z88. Y cuando las pilas necesitan ser cambiadas, un condensador especial (o el adaptador de corriente) preserva los datos.

El Z88 no solamente es capaz de hacer ruido, sino que también posee capacidades gráficas. Puede utilizar una zona de 256x64 pixel (los juegos serán en CINEMASCOPE).

El Z88 es bastante sólido:

" Resistió lluvia, arena, granizo, nieve y viento en Israel, viajando en un 747, autobuses Israelitas y Egipcios e, incluso en la grupa de un camello.

En mi opinión, gana todos los premios de peso, tamaño, resistencia y facilidad de manejo tanto en el campo como en la ciudad"

Charles Levy
PC Business World
Marzo 1988.

Como os podeis imaginar, la critica es muy favorable debido en parte a que no interfiere el mercado PC, ya que es un producto complementario (aunque habria que ponerlo en duda, porque es un ordenador personal capaz de darle mil y una vuelta a un IBM PC) .

Es muy posible que el paquete de aplicaciones en ROM sea el adecuado para cubrir las necesidades de la mayoria de los usuarios potenciales (aunque la explosión de programas comerciales no hace ni 6 meses que ha comenzado).

Las aplicaciones trabajan de una manera parecida a lo que estamos acostumbrados con el QL. Podemos tener varias tareas a la vez, pero solamente una corriendo en cualquier momento (en el QL podemos tener varias tareas trabajando con una prioriodad asignada e, incluso alguna suspendida); estando todas las demás suspendidas en el lugar que se dejaron. En el QL la multitarea es real, en el Z88 no existe la multitarea como tal, sino varias tareas en memoria suspendidas esperando ser utilizadas; pudiendo tener cada una su propio nombre aunque sea un programa BASIC u otra tarea PIPEDREAM. De todas formas el sistema operativo OZ es una maravilla comparado con el MS-DOS.

Algunos detalles de interés :

Solamente se puede usar el slot 3 para cartuchos Eprom.

Los cartuchos Eprom por su precio creo que no van a tener mucho éxito (hay almacenamientos alternativos: Cassette, disco y en otro ordenador). pero los programas comerciales tienen que usarlos con el inconveniente de que encarece notablemente el precio final del software.

La versión Española posee todos los caracteres Españoles, y además tiene acceso a todos los caracteres Ingleses y Europeos. Pero los programas suministrados en ROM no están traducidos.

Se suministran dos manuales : uno en Inglés y el otro en Español (el original en Inglés es más completo).

El BBC BASIC suministrado debe tratarse de una versión muy evolucionada del BBC BASIC de 1982, pues parece una mezcla de Superbasic y BASIC Microsoft. Posee procedimientos, funciones y repeat. No dispone de los comandos gráficos y sonido.

S. Merino, Fuengirola, 2/5/89.

EN EL PROXIMO NUMERO

INSTALANDO LA IMPRESORA

---

C.U.Q. SOFT

---

Por fin tenemos lista de programas. Todos los programas han sido racionalmente renombrados, y alterados sus "boots" en consecuencia, de modo que sean fácilmente manipulables mediante WDIR, WCOPY... y de ese modo acelerar y facilitar la labor de copiarlos a quien los solicite, así como posibilitar su ordenado almacenamiento.

NORMAS PARA CEDER PROGRAMAS A LA LIBRERIA

Se ruega que, ¡por favor!, a la hora de ceder programas a la librería, se respeten las siguientes normas:

1.-Poner a cada fichero una extensión adecuada que indique su tipo, por ejemplo:

- _bas si es un programa en SuperBASIC
- _exe o _task si es un programa EXECutable
- _bin si es código máquina no EXECutable
- _txt o _asc si es un fichero de código ASCII
- _fth si es un texto fuente es Forth
- _scr si es una pantalla, o un fichero fuente del Forth de Computer One
- ...etcétera

2.-Poner a todos los ficheros de un mismo programa un mismo nombre inicial, que no coincida con alguna de las referencias de los programas ya existentes. Si entre los ficheros de un programa existe un fichero _doc del Quill, o de otro paquete de Psion, el nombre inicial común a todos los ficheros no podrá, por tanto, ser más largo de 8 caracteres.

3.-No emplear en los nombres de los ficheros caracteres que no sean aceptados directamente, es decir, que no sea necesario encerrar el nombre entre comillas para manipular el fichero.

4.-Deben enviarse, como las colaboraciones escritas, a: Salvador Merino, Carretera de Cádiz (Cerámicas Mary), 29640 FUENGIROLA (Torreblanca del Sol), Málaga.

LISTA DE PROGRAMAS DISPONIBLES

(De momento no se incluye relación de ficheros, ni sectores ocupados.)

| CUQ | Referencia | Autor | Descripción breve |
|---|---|---|---|
| 1 | sfthcalc | Salvador Merino | Calculadora EXECutable (RPN) |
| 1 | ratón | J. C. de Prada | Tkit para SMouse desde BASIC |
| 1 | rr | J. C. de Prada | Juego retrato robot |
| 2 | sísiforth | Marcos Cruz | Sistemas Expertos en Forth |
| 4 | fthmat64 | Salvador Merino | Matem. de 64 bits en Forth |
| 5 | exchangeador | Agustín Gallego | Intercambia |
| 5 | traductor | J. C. de Prada | Traduce códigos IBM <-> QL |
| 6 | cuatro_en_raya | Luis Valero | Juego 4 en raya, 2 jugadores |
| 6 | ahorcado | Luis Valero | Juego del ahorcado, sin graf. |
| 6 | laberinto | Luis Valero | Juego salir de laberinto 3D |
| 6 | amplia_gráficos | Luis Valero | Retocar pantallas, modo 4 |
| 6 | copy_grises | Luis Valero | Rutina para hard-copy |
| 6 | sonidos | Luis Valero | Ejemplos al azar de BEEPs |
| 6 | errores | Rafael Candau | Para cambiar m. de error QL |
| 6 | cuentas | Rafael Candau | Plantilla _aba doméstica |
| 6 | clock | Luis Valero | Reloj EXECutable |
| 7 | printer_proc | Anónimo | Proc. para config. impresora |
| 7 | tenis | Agustín Gallego | Juego Ping-Pong, 2 jugadores |

(Faltan un par de ellos, que están pendientes de que se aclare el objeto de alguno de sus ficheros.)

Los programas que se desee pueden solicitarse, bien individualmente por su nombre de referencia, o bien por el número de CUQ al que corresponden a: Marcos Cruz, Acacias 44, 28023 Madrid; adjuntando sobre (suficientemente) franqueado y con la dirección escrita, así como disco o microdrive/s formateado/s.

Marcos Cruz, 4/4/1989

Junto con este boletín, se incluyen los siguientes programas:

- Tenis_plus_bas.- Versión ampliada del anterior programa TENIS por Marcos Cruz.
- Lister.- Para listar programas PASCAL. Varias versiones: Turbo, Qliberator,... Autor: José F. Rojas
- Trans-base.- Utilidad Forth para cambios de bases sin esfuerzo. Autor: J. C. de Prada.

-----------------------------------------------------------------------

EN EL PROXIMO NUMERO...

-----------------------------------------------------------------------

- Nuevo programa para pasar pantallas del Spectrum al QL.
- Comentario sobre el founted89.
- Traducción del manual del founted89.
- Practicando con el Superforth (XIII).
- Segunda entrega Compilador Cobol IEC-SOFT.
- Traducción manual Toolkit II (segunda entrega)
- Traducción manual Qram
- Traducción manual drivers (Tony Tebby)
- Traducción manual QIMI
- Traducción manual Qliberator
- Y muchas cosillas más (incluido programas) de cuyo nombre no deseo acordarme.